AD-A090 234   COLORADO STATE UNIV FORT COLLINS DEPT OF ATMOSPHERI--ETC F/G 4/2
DIGITAL IMAGING RESEARCH FOR METEOROLOGICAL APPLICATIONS.(U)
AUG 80 T A BRUBAKER, T H VONDER HAAR   DAAG29-76-G-0324
UNCLASSIFIED   ARO-13971.7-GS   NL

1 OF 2
AD
A090234

LEVEL (10) ARO 13971.7-GS

SUPERSEDES PREVIOUS RPT.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
(18) ARO (19) 13971.7-GS

2. GOVT ACCESSION NO.
AD-A090 234

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(16) DIGITAL IMAGING RESEARCH FOR METEOROLOGICAL APPLICATIONS.

5. TYPE OF REPORT & PERIOD COVERED
(9) Final Report.
1 Sept. 1976 - 31 July 1979,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Thomas A. Brubaker, Thomas H. Vonder Haar,
Gregory C. Smith, and Marvin Brown (15)

8. CONTRACT OR GRANT NUMBER(s)
DAAG29-76-G-0324

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Atmospheric Science Department
Colorado State University
Fort Collins, CO 80523 (12) 14

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(11) 21 Aug 80

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
Post Office Box 12211
Research Triangle Park, NC 27709

12. REPORT DATE
2/22/80, revised 8/19/80

13. NUMBER OF PAGES
12

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

A

18. SUPPLEMENTARY NOTES
The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official department of the Army position, policy, or decision unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Meteorology, Atmospheric Science, Electrical Engineering, Satellite Data, Radar Data, Computer Imaging

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Planning for this joint research on digital imaging by electrical engineers and atmospheric scientists began in early 1976. At that time the use of digital solid state memory under small computer controlto refresh a CRT to produce a "digital image" was in its infancy. ~~Our~~ research focused on the design, procurement, test, demonstration and meteorological research use of the new system. It was very successful and has received national and international acclaim. Several papers were published under sponsorship

088310

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

80 10 9 001

of this grant and are listed in an Appendix. This Final Report includes complete details of the research on both hardware and software aspects of the system. This research has already lead to the design and research use of the next generation of digital imaging systems.

TABLE OF CONTENTS

INTRODUCTION

It is important to note that the engineering design, procurement test and meteorological application of results of this research have recieved national and international acclaim. Hundreds of scientists and engineers have personally visited the Colorado State University specifically to view, discuss and carryaway information about the results of the research on"ADVISAR".

The research that was performed consisted of:

1. Design and development of a prototype digital image refresh system for use with quantitative satellite research and application.

2. Design and development of a mini-computer interface suitable for a field refresh unit.

3. Development and integration of software in the meteorological applications area.

4. Use of the new systems to carryout satellite/weather/radar research on clouds.

As a direct result of the research, several scientific papers and presentations were completed. These are catalogued in Appendix A. Also several reports describing the hardware have been developed and these are referred to where needed. Each of the above categories is discussed in detail below.

1. Prototype Digital Image Refresh System

The digital imaging system named ADVISAR (All Digital Video Imaging System for Atmospheric Research) was designed during the first phase of the work. The design was then subcontracted to INTEL Corporation for construction and the system was on-line during the summer of 1977. This phase of work was funded in part by Colorado State University as a matching effort and demonstrated cooperation with various research groups and industry to develop a state of the art tool for meteorological research.

The ADVISAR system consists of eight frames of solid state memory with each frame consisting of 512 x 512 - 8 bit pixels. In the initial phase the system was run under control of an HP-2100 computer. Since then the control has been transferred to a DEC PDP-11/60 as described in the next section. The most important feature of ADVISAR is its capability to rapidly display eight images therby allowing the meteorologist to visualize the temporal aspects of satellite images. A detailed description of the system has been published and the report is included in Appendix B.

After completion of ADVISAR, the acquiring of good quality satellite data began to be a problem. We recognized that if we were to do long term research, a groudstation at CSU would be necessary. This goal has now been achieved. We have today a total acquisition and display system centered on the concept of using a man-in-the-loop to guide the processing and display of the meteorological imagery. The system software provides the user with the ability to effectively use the system. This is described in greater detail in Section 3.

At this time there are a variety of vendors who market image systems and further development work in the university is not needed. These systems have evolved because of inexpensive solid-state memory. However, the total systems aspects of meteorological image processing still needs work if modern integrated circuit technology is to be incorporated into further applications.

2. New ADVISAR Interface

During the last phase of the project a new interface to the DEC PDP-11/60 computer was designed, constructed and placed into operation. This has allowed the ADVISAR system to be controlled by the latest generation of mini-computers thus providing vastly improved capability. With the groundstation, the total system configuration now appears as shown in Figure 1.

A custom interface, based on a modified general purpose DMA (Direct Memory Access) foundation module, now connects CSU's ADVISAR memory system to the DRSGS (Direct Readout Satellite Ground Station) PDP-11/60 UNIBUS (Figure 2). In addition to the DMA module, the interface includes user logic assembled on a seperate quad wire wrap card. Zvi Barak designed and assembled user logic; for detals see Appendix C.

From Figure 3, fixed logic on the DMA foundation module includes UNIBUS drivers and receivers, an address decoder, four registers (input data, output data, word count (WCR), and the bus address (BAR), a multiplexer for selecting and gating register information onto the UNIBUS, interrupt logic, and a bus-master controller. Several modifications enhance capabilities of the DMA module: additional logic permits byte oriented (rather than standard work oriented) data transfers between interface and peripheral device; expanded address decoding supports an off-board command channel buffer register; and a WCR all-ones detector controls DMA data transfer completion.

Figure 4 presents, in block diagram form, wire wrapped ADVISAR interface user logic. Board components include a control and status register (CSR), a command channel buffer register, an output data control logic for communications, with the modified DMA module and ADVISAR. Users establish ADVISAR system staus via commands sent over control channel lines CBIT00 - CBIT15; pulse DCCC initiates command transfer. To prevent flickering or flashing images, some command transfers take place during verticle blanking periods only (N Imsec every 1/60 second); CCFLAG generates an 11/60 interrupt whenever ADVISAR videotiming signals initiate display monitor vertical retrace operations. After establishing proper status, the interface controls transfer of data bytes either to (WEAll - WDA07) or from (RDA00 - RDA07) ADVISAR image or enhancement table memory. Timing pulse DCDC initiates data transfer and, subsequently, DCFLAG signal transfer completion. The detailed design is given in Appendix C.

3. Software Development for Meteorological Application

The software is the product that interfaces the systen and the user. Recognizing that the data collection and ADVISAR systems would be used by a variety of users having computer skills ranging from none to good, a keyed system design was chosen. The system had to be modular so that new applications software could be easily inserted while maintaining system integrity and operation. Man/machine communication had to be easy and therefore included the use of well designed cursors to simplify the location of specific regions of a given image.

Once achieved, this resulted in a variety of scientific products. Seperate radar and satellite observations were composited on the ADVISAR to monitor convective development, the organization of mesoscale features with respect to the synoptic scale, severe storm signatures and convective and stratiform precipitation during the fields program of the bureau of Reclamation's High Plains Experiment (HIPLEX).

Extensive documetation of the computer software is provided in Appendix D.

4. Summary of Meteorological Research Accomplishments Using the ADVISAR (All Digital Vedeo Imaging System for Atmospheric Research)

Smith and Reynolds, 1978, have developed procedures whereby remotely sensed radar and satellite observations can be composited in common map projections and displayed using an interactive image processing system (ADVISAR) which uses high fidelity video display technology.

The 5 cm SWR-75 digital radar data and GOES-East visible satellite data are prepared separately and stored in the digital video refresh memories of ADVISAR. The two separate memories are then composited on the color television monitor using seperate color input channels. As the magnitudes and respective ratios of the digital equivalence function values for the radar dbz and satellite visible intensity vary the corresponding color enhancement signatures change. Using this same technique a third channel, the infrared, has been added to the compositing scheme.

The digitally based compositing technique provides a significant improvement over conventionally presented radar and satellite data by reducing the amount of information needed to be assimilated by the forecaster. More importantly it provides an immediate focus to the forecaster with respect to the satellite information. Since the radar data alerts a forecaster to the potential for severe weather, the satellite observed growth processes available to the forecaster with this technique can now be monitored directly in conjuction with echo growth. This will allow the forecaster to concentrate on the actual developing weather phenomena rather than the mechanics of orientation. Ultimately it is hoped that a standardized forecast product using radar-satellite composites will be developed.

Reynolds and Morris, 1978, have demonstrated the usefulness of the ADVISAR system in the analysis of digital satellite data for operational seed, no seed decision making with regard to cold orographic clouds set up by the Sierra Nevada mountain barrier in East-central California. In support of the Sierra Cooperative Program of the Water and Power Service they examined digital infrared imagery for several case study days to determine cloud top temperatures over the Sierra Program target region.

Previous research has shown that the cloud's top temperature is related to its seedability. Specifically, clouds with teperatures between 10 to 25°C have deficient ice nuclei for efficient precipitation and therefore such clouds, if seeded, could increase their potential precipitation significantly.

The usefulness of the digital satellite imagery displayed on the ADVISAR system lies in its ability to sense and graphically display area-averaged cloud top temperatures at short time intervals. This area-averaged temperature may give a more representative value of cloud top temperature than the point value obtained from a sounding due to the displacement of the sounding from the area of interest and the spatial variation of cloud tops where embedded convection may exist.

The case studies examined revealed a good comparison between satellite/ADVISAR derived temperatures and cloud truth data acquired from aircraft. These results have allowed CSU, with its Direct Readout Satellite Ground Station, to initiate real-time satellite support for decision making purposes during subsequent Sierra Cooperative Program field seasons.

Reynolds, 1978, has used the ADVISAR to make rainfall estimates over a low density rain gauge network of the Water and Power Services HIPLEX (High Plains Experiment) field site near Miles City, Montana. This study was similar in nature to that by Smith and Reynolds, 1978, except that in addition to digital radar and digital geosynchronous satellite data to estimate cloud heights and temperatures, data from a high density rain gauge network at the core of the field site was also used in making estimates of precipitation in the much larger low density rain gauge network surrounding the core area.

Results of the study showed that digital satellite data compared favorably with the radar and rain gauge data. A bi-spectral approach showed possibilities for determining rain-no rain events but did not appear to quantify the rain ammounts. Having digital data available in the ADVISAR system allowed mathematical transformations of each spectral data set as well as a combination of the visible and infrared data into a single bi-spectral image. A preliminary technique was tested, inverting the IR data [i.e. cold-low count, warm-high count (as it would be truly sensed by the detector)] and subtracting it from the visible data. This resulted in a bi-spectral image whose values can be monitored with time to determine a relationship to rainfall.

A few further results showed that the infrared data, although underestimating the maximum cloud radar height, did show the relative changes in top height which related well to rainfall at the surface. Maximum rainfall appears to occur before the coldest cloud-top temperature is reached. Vertical wind shear appears to be important in determining rainfall efficiency and should be considered in a satellite rainfall method. The visible reflected brightness is an earlier indicator of rainfall, although its magnitude is affected strongly by geometric factors that must be accounted for before a bi-spectral rainfall study can be performed.

Finally, Reynolds, Brown, Smith and Vonder Haar, 1978, used ADVISAR in studying cloud type separation by spectral differencing of image pairs. The raw infrared and visible counts were normalized to an 8-bit scale (0-255) but were not converted to radiometrically linear units. The difference image was also rescaled to the 0-255 range. Therefore all three images (infrared, visible, difference) utilized the complete dynamic range of the ADVISAR video refresh memories. Clouds that were cold and bright (high IR count, high VIS count) produced a difference signature in the gray or medium values. Cold and dim regions (high IR count, low VIS count) appeared as the very lowest values in the difference image, or black. By adding image sequencing to the procedure it was possible to remove any remaining ambiguities in the separation of clouds from snowfields. Simple subtraction of images using objective, quantitative threshold values resulted in a technique to identify cirrus clouds.

[See Appendix A for a bibliography of additional scientific reports and publications sponsered by this research grant.]

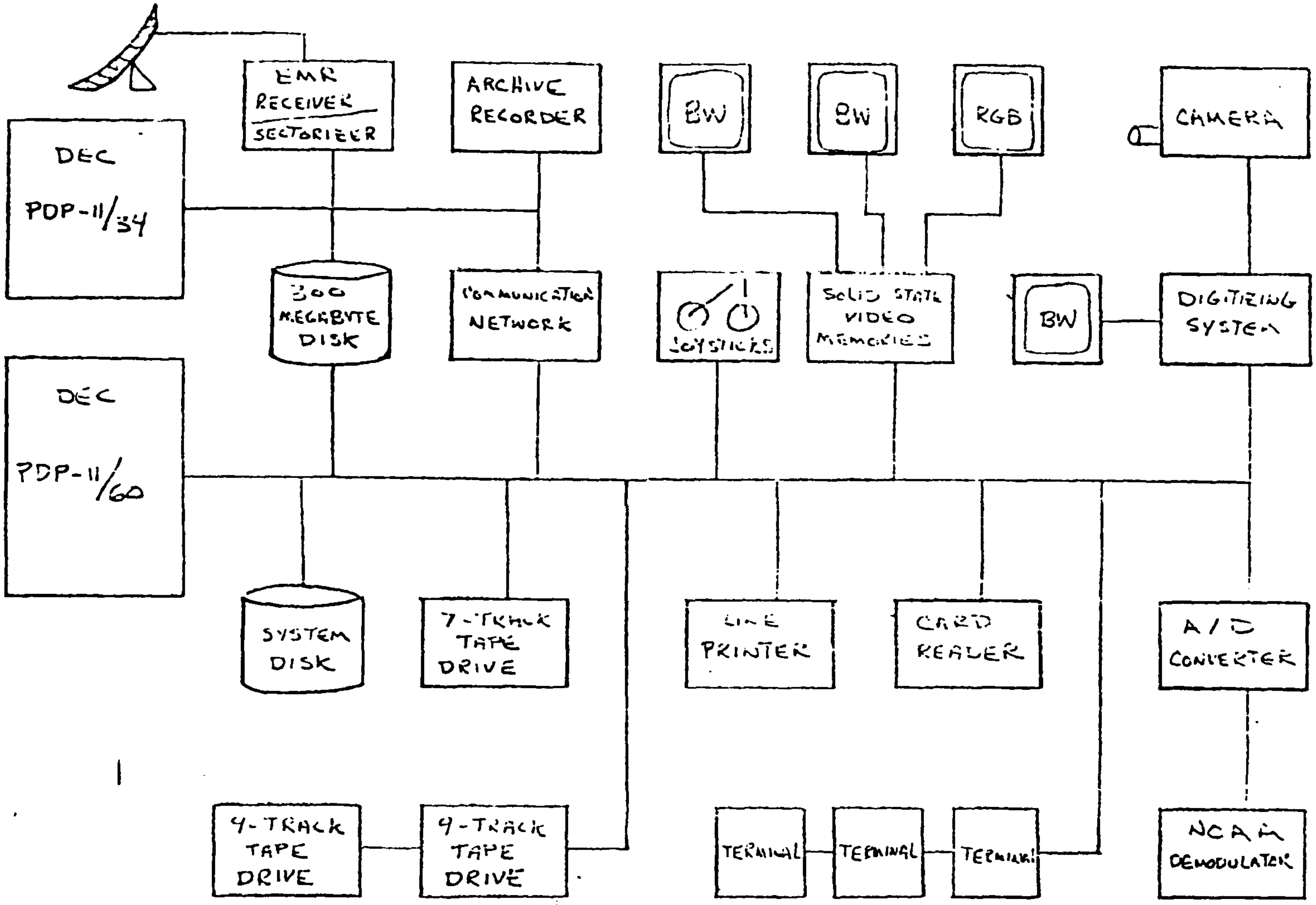


FIGURE 1. PRESENT SYSTEM BLOCK DIAGRAM

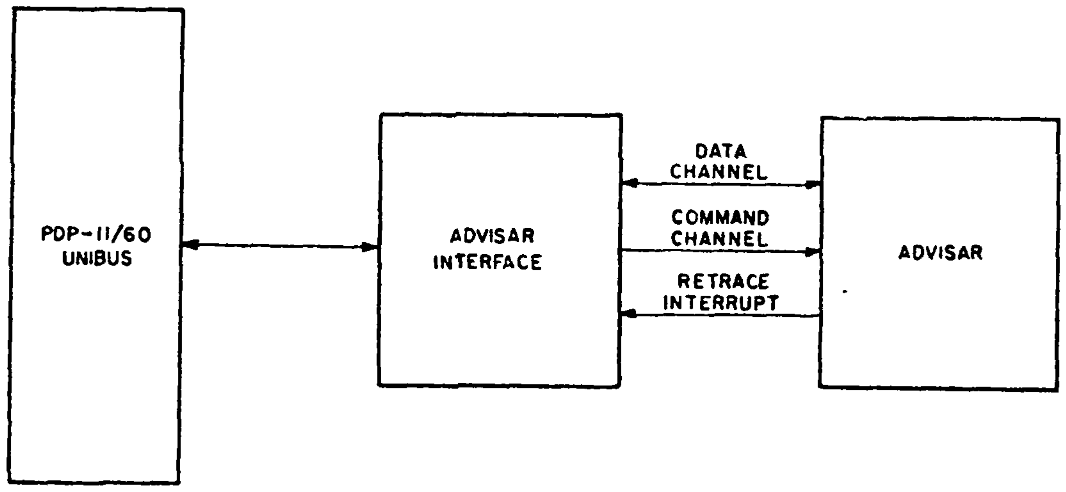


Figure 2. ADVISAR Interface

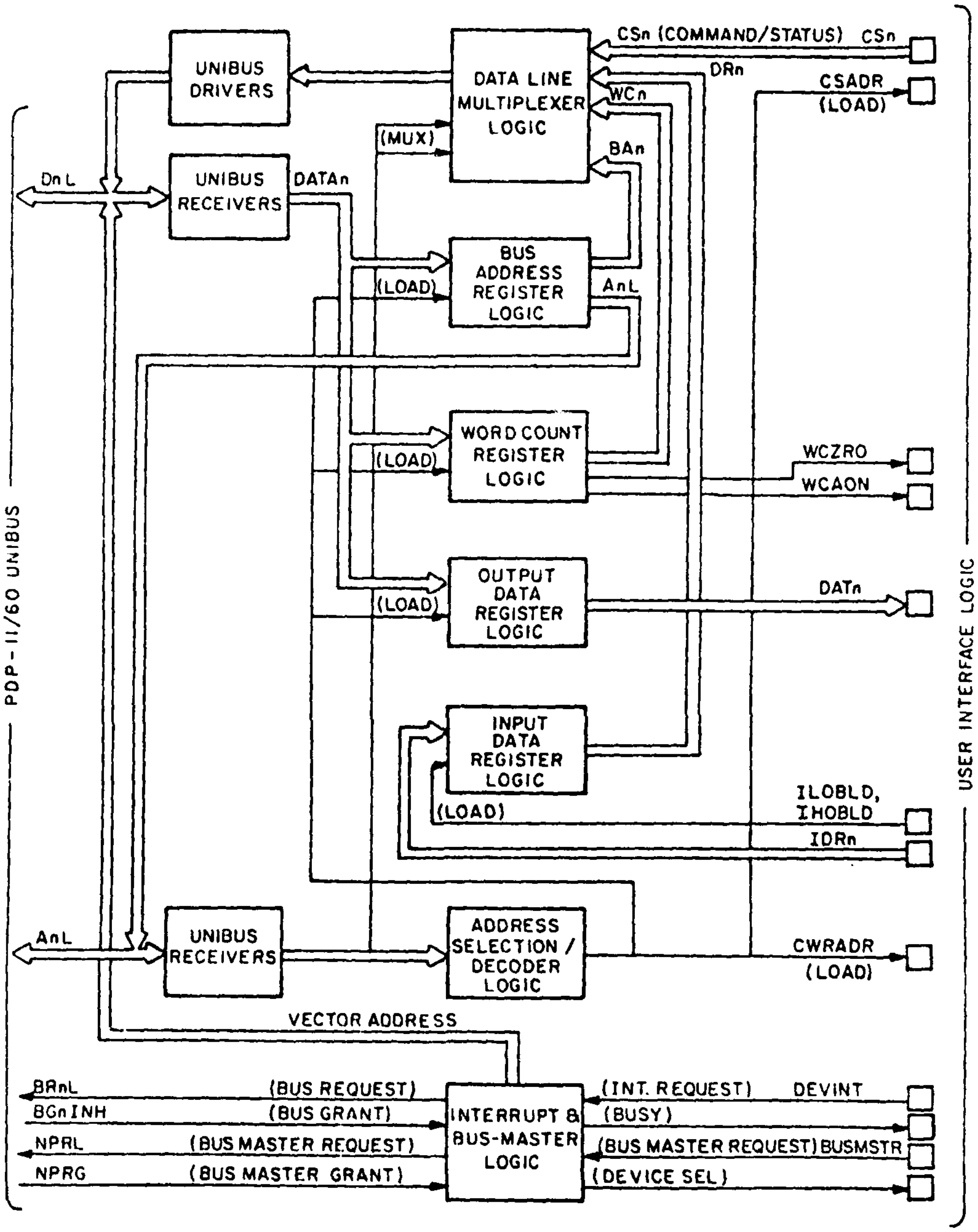


Figure 3. Modified General Purpose DMA Foundation Module

Appendix A

SCIENTIFIC PAPERS AND PUBLICATIONS SPONSORED BY GRANT DAAG29-76-G-0324

Andrews, M. and R. Fitch, 1979: "A Digital Image System for Atmospheric Research". Accepted for publication in the International Journal of Computers and Electrical Engineering, Vol. 5, No. 4, April.

Brown, M. L., 1978: "Digital-Video Manipulation of Satellite Imagery". Preprint of the Fourth Symposium on Met. Observations and Instrumentation, Denver, CO, April 10-14.

DeMasters, D. and M. Andrews, 1978: "Best Fit Edge Detection for Meteorological Data". Preprint of the Society of Photo-Optical Instrumentation Engineers, 22nd Annual Technical Symposium, San Diego, CA, August 29-31.

✓Haass, Uwe L. and T. A. Brubaker, 1979: "Estimation of Lateral and Rotational Cloud Displacement from Satellite Pictures". Workshop on Computer Analysis of Time-Varying Imagery, Philadelphia, PA, April.

Reynolds, D. W., 1978: "An Intensive Analysis of Digital Radar, Raingage and Digital Satellite Data for a Convective Cloud System on the High Plains of Montana". Preprints of the 18th Conference on Radar Meteorology of the AMS, Atlanta, Georgia, March 23-31.

Reynolds, D. W. and K. R. Morris, 1978: "Use of the ADVISAR in the Analysis of Digital Satellite Data for Operational Seed no Seed Decision Making". Preprints of the Fourth Symposium on Meteor. Observations and Instrumentation of the AMS, Denver, CO, April 10-14.

Reynolds, D. W., M. Brown, E. Smith and T. H. VonderHaar, 1979: "Cloud Type Separation by Spectral Differencing of Image Pairs", Mon. Wea. Rev., Vol. 106, No. 8, August.

Smith, E. A. and S. Q. Kidder, 1978: "A Multi-spectral Satellite Approach to Rainfall Estimates". Paper presented at the 18th Conference on Radar Meteorology, Atlanta, Georgia, March 23-31, 48 pp.

Smith, E. A. and D. W. Reynolds, 1978: "The Generation and Display of Digital Radar-Satellite Composites using Analytic Mapping Techniques and Solid State Video Refresh Technology". Preprints of the Conference on Weather Forecasting and Analysis and Aviation Meteorology, Silver Springs, MD, October 16-19.

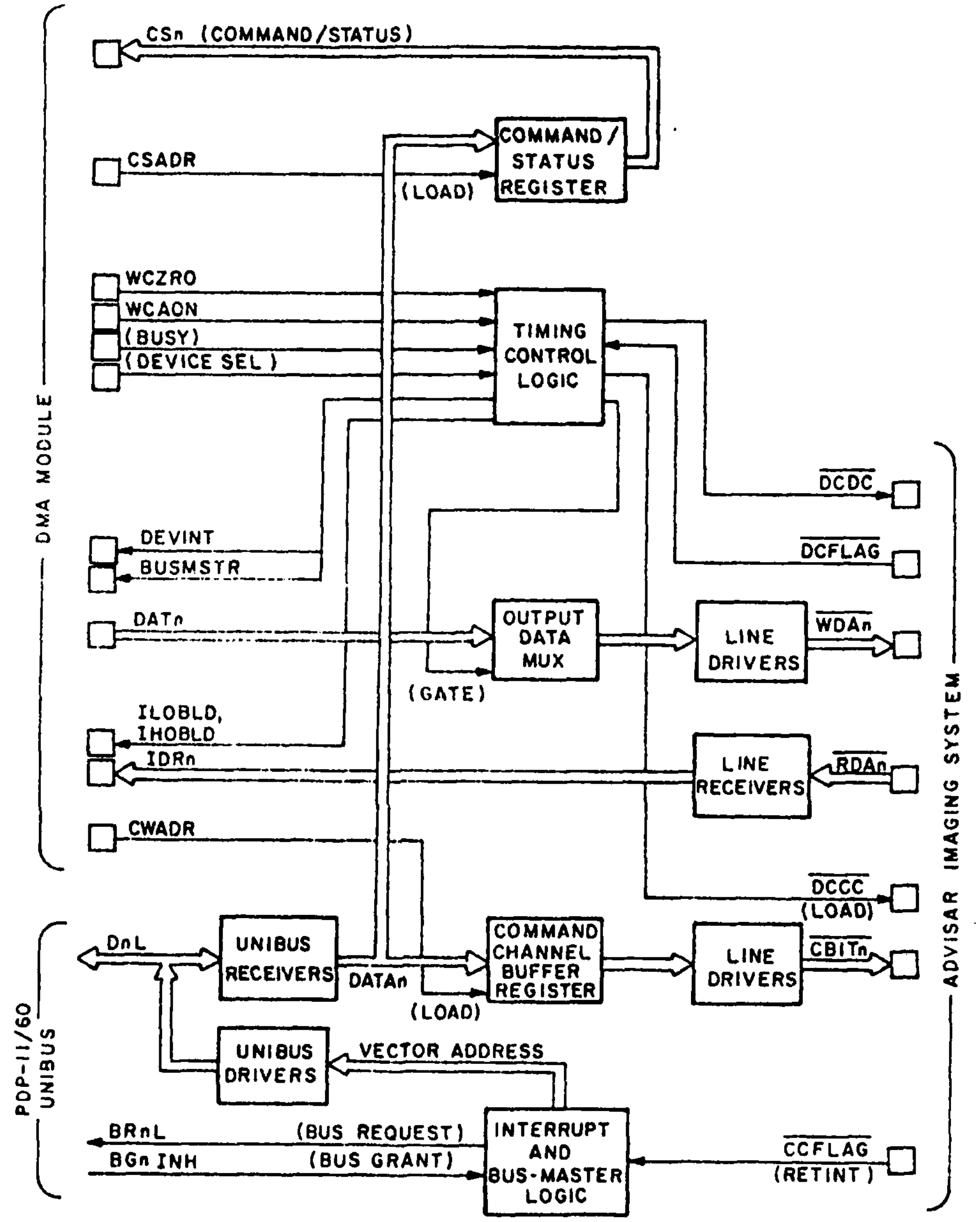


Figure 4. User Interface Logic

APPENDIX B

CAEE 185--P. 1.

CAMERA READY COPY WILL BE
SENT WHEN AVAILABLE

A DIGITAL IMAGE SYSTEM FOR ATMOSPHERIC RESEARCH[†]

MICHAEL ANDREWS[‡] and ROBERT FITCH

Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A.

(Received 5 April 1978; received for publication 21 September 1978)

Abstract—This paper describes the architecture of a medium scale digital image processing system developed as a research tool for analysis of meteorological data. The system is also being used for research on efficient image processing systems. Four qualitative performance measures for any image processor are introduced with specific application to the present machine. Preliminary results with noise reduction algorithms in satellite data are presented. Lastly, the versatility of the machine as a test bed for architectural studies of the computational structure of image processors with a microprogrammable control unit is discussed.

1. INTRODUCTION

Our stratosphere, residence of the world's weather, has largely been observed beneath the clouds. Today, however, the launch of the fifth Synchronous Meteorological Satellite (SMS-5) will herald a new dimension of contiguous weather observation from above when the five satellites will each scan 20% of the earth and transmit on a downlink visible and IR intensity data via binary-coded pixels every half-hour[1]. Unfortunately, at present such voluminous data is received at transmission rates which rapidly become archived, mainly due to primitive real-time data processing systems. Only recently have we been able to rapidly use data for current analysis. In addition, we now can interpret complex weather behavior at mesoscale levels, in severe storms and hemispheric cloud motions because of recent advances in digital display systems, one of which is described in this paper[2]. These new analytical tools greatly assist in the current pursuit[3-12] for reliable and cogent weather prediction. Typically, such systems employ digital imaging techniques which have (a) high resolution, (b) stable image displays for microscopic analysis, (c) high speed image sequencing for temporal studies, (d) image enhancement for highlighting specific intensity regions (contour banding), and (e) real-time cursors, zoom, translate, and rotate for interactive capability. The need for such requirements and quantitative measures for evaluation have been proposed by [13, 14] and discussed in [15] and will not be repeated here.

Of considerable interest to atmospheric research is the ability to replicate weather maps transmitted in real-time from satellites and portrayed as high fidelity video scenes coupled with flexible graphic overlays for political and geographical registration. However, in the past, these techniques have required manual generation of picture composition by photomosaics necessitating extensive wet chemistry support (for developing, enlarging, etc.). Now, we can employ recent advances in solid state electronic technology (semiconductor memory, microprocessors and high resolution video monitors) which virtually automate all phases of the weather map generation. Other attractive features, now feasible, include enhancing invisible scenes (hot air masses), etching details by pseudocolor and overlaying topographical as well as geographical features of landscape.

Such capability which exists to some extent in all large scale systems[6-9] relies heavily upon menu-driven software packages desirable for reducing inexperienced-user training. In essence, several modular routines are provided for the user via either a video screen tabloid entry[6], trackball/cursor[9], or special function keys[7, 8]. Such software coupled with annotated cursor, zoom, translate and rotate capabilities in the above systems as well as in small systems[10, 11] enhances the human manipulative skills in a tight man-machine interactive loop. However, to ensure rapid analysis and preserve continuity of thought, quick machine response

[†] This work was supported by a grant from Army Research Office under grant No. DAAG29-76-G0324.

[‡] Address all correspondence to Michael Andrews, Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A.

Smith, G. W., Loendorf, U. Haass, J. Snowden and T. Brubaker, 1979: "Analysis of Interpolation Methods for Meteorological Satellite Images". IEEE Conference on Space Instrumentation for Atmos. Observations, El Paso, Texas, April.

Vonder Haar, T. H. and E. A. Smith, 1979: "Fourth Generation Display Systems and the Multiple Platform Problem". Presented at the IEEE Conference on Space Instrumentation for Atmos. Observations, El Paso, Texas, April.

APPENDIX C

JOYSTICKS AND KEYBOARD UNIT
FOR ADVISAR

Built for use in the Department of Atmospheric Science

Designed and Written by

Zvi Barak
Department of Electrical Engineering
Colorado State University

November 1978

INDEX

APPENDICES

PREFACE

The system described hereby was designed for replacement of a similar existing unit, which is now employed by the ADVISAR (All Digital Video Imaging System for Atmospheric Research), a system which is now in use in the Department of Atmospheric Science, Colorado State University. In the first place, redesign of the unit was required because of several problems that appeared to result from poor performance of the existing unit. Later on it was made clear that the problem was not with the joystick unit, but rather in the interfacing circuitry. Anyhow, since the system was intended to be switched over from utilizing an HP-2100 computer to the DEC PDP-11/60 computer, and meanwhile, additional requirements from the unit came up, resulting from the accumulating experience gained by the system users, a decision was made to redesign and rebuild the unit according to the new requirements. Building an additional unit also results in avoiding ordinary system operating disturbances during the transient period of replacing the computer.

This manual describes this unit in detail as well as describing external subsystems which are not located within the unit cabinet, as interfacing facilities or test equipment, for instance. Operation and test and alignment procedures are also described.

1.0 SYSTEM BLOCK DIAGRAM

1.1 Subsystem Division:

The unit consists of 3 main parts: The DC subsystem, the logic subsystem and the power supplies. All those parts are accommodated in the unit itself. External subsystems, as the interfaces for example, that appear in the block diagram in dashed lines, or the test, box, will be described in other following chapters.

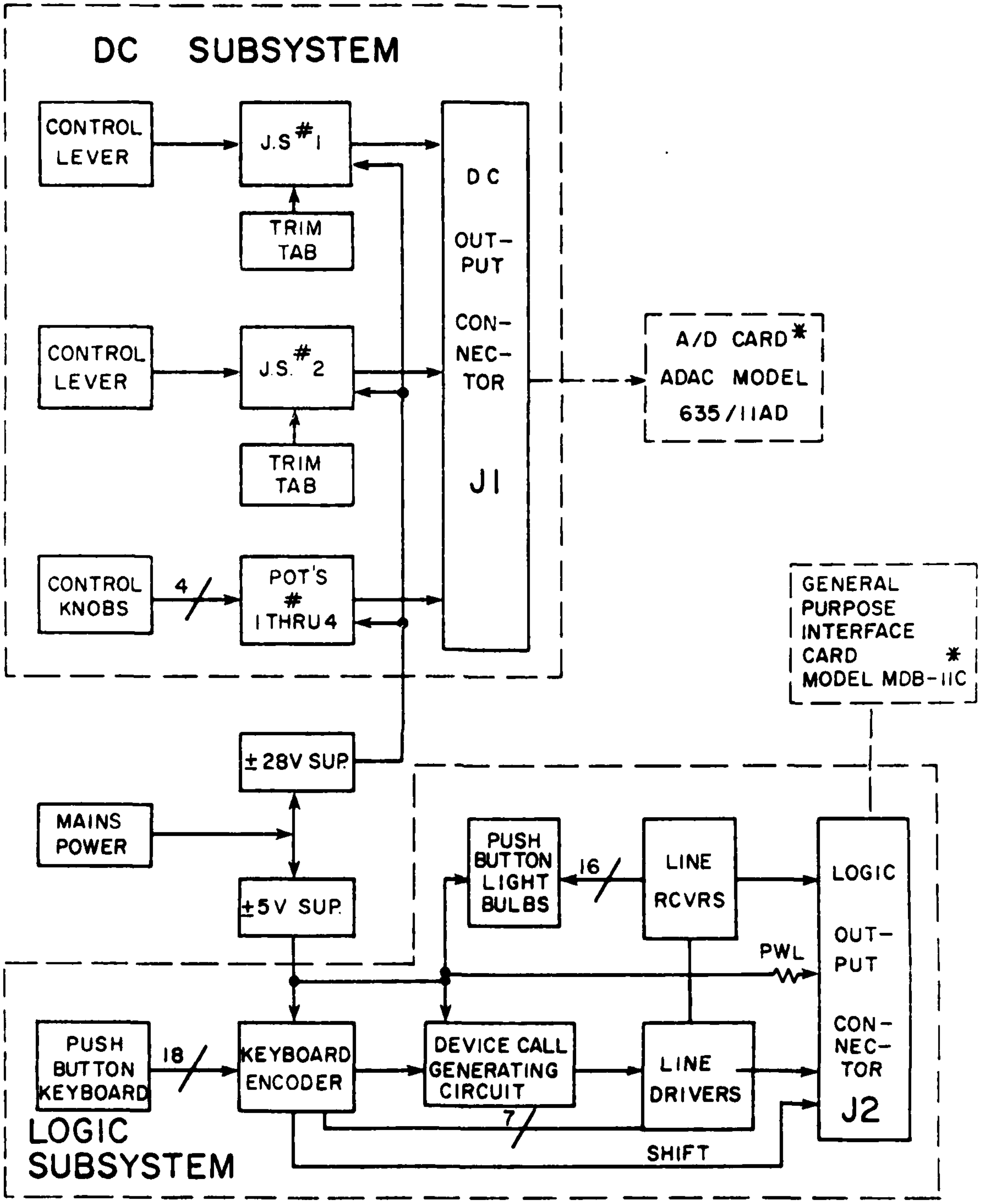
1.2 DC Subsystem:

This subsystem includes the circuitry for the joysticks and for the manual data potentiometers. Each joystick employs two 5 kilohm potentiometers, one per each axis. In this way, the DC subsystem includes as many as 8 DC output channels. Those signals are routed via the DC output connector to the Analog-to-Digital Converter, which is a multiple-channel interface card, plugged in directly into the PDP-11/60 computer mainframe.

1.3 Logic Subsystem:

The main parts of this subsystem are the 16 push-button keyboard and the encoding circuits, where the keyboard data are encoded into a 6-bits combination. The keyboard includes also 2 other keys: END key, with a yellow lens, and SHIFT key, with an orange lens, marked with ⬆. The first 16 keys have white lenses. Most of the output signals are buffered by line-drivers, except SHIFT, which is driven directly through the push-button contacts.

A 100μSec pulse is sent from the logic to the interface card (DVC-Device Call), that strobes the data bits into the input register, which is located in the interface card, type MDB-11C, and also initiates a bus-request that results in an interrupt call.



* DASHED LINES BLOCKS DESCRIBE SUBUNITS NOT LOCATED WITHIN THE CABINET ITSELF.
OTHER DASHED LINES MARK SUBSYSTEMS WITHIN CABINET.

Fig. 1.1. JOYSTICK AND KEYBOARD CABINET/BLOCK DIAGRAM

The 16 light bulbs in the keyboard push-buttons are controlled entirely by the program, and the computer issues, from time to time according to the software, a 16-bits word which is recorded in the MDB interface output register. Every bit corresponds to one bulb, the bit DA00 (least significant) corresponding to bulb A, and the bit DA15 (most significant) corresponding to bulb S. The signals are received in the cabinet by line receivers and routed into driving transistors that drive the bulbs lighted.

1.4 Output Connectors

Since the two subsystems signals are routed to two separate interface cards, they are also transferred via two separate output connectors. J1 is the DC output connector (24 pins) and J2 is the logic output connector (36 pins).

1.5 Power Supplies

Three power supplies are necessary for the unit's proper operation. A +28V and a -28V power supplies are required for the DC subsystem operation, and a +5V power supply is required for the logic circuitry, which consists mostly on TTL IC's. The current consumed by this latter power supply load is relatively high, about 1.5A, since the +5V voltage is also required for illumination of the push-button bulbs.

2.0 SYSTEM DESCRIPTION

2.1 Operating Instructions:

The cabinet accommodates 2 joysticks, four 5-kilohm potentiometers and an 18-keys keyboard. The unit is packed in a slant-panel type DSSF-7006 Buckeye manufactured cabinet.

2.1.1 Joysticks:

Two joysticks are mounted on the slant-front panel of the unit. The joysticks are both of type 521-2513, with adjustable friction hold facility for each axis, manufactured by Measurement Systems, Inc. Joysticks are also equipped with a trim-tab on each axis to facilitate output voltage centering alignment. The left hand joystick (#1) is used for assigning cursor location on the system monitor, while the right hand joystick (#2) is used for assigning cursor dimensions (height and width). For additional data about the joysticks, see Appendix A.

2.1.2 Potentiometers

Four (4) identical panel controlled 5-kilohms potentiometers are mounted on cabinet access panel, on its part which is further from the operator, leaving room for the keyboard on the part closer to the operator. Those pots are used for analog data manual assignment. All joysticks and pots outputs, concluding total of 8 DC channels of data, are converted into digital data by an 8-channel A/D converter, which is plugged directly into computer card-cage, thus interfaced to the UNIBUS.

2.1.3 Keyboard

The keyboard consists of 16-bit main keys and 2 auxiliary keys. The 16-bit keys are marked A through S, skipping the letters I, O

Fig. 2.1. FRONT VIEW OF THE CABINET

and Q so that they will not be confused with one, zero or with each other. The 16 keys represent a keyboard on which messages can be keyed in. The keys can be illuminated, and their illumination is entirely program controlled. The program only can decide which key will be illuminated and when. Every press on a key is encoded into a 6-bit word according to the following table (6 bits word represented in octal code).

A	22	J	11
B	01	K	10
C	02	L	20
D	03	M	30
E	04	N	40
F	05	P	50
G	06	R	60
H	07	S	70

Table 2.1 Keyboard Code

The six bits are read into the six least significant data lines of the UNIBUS.

Besides those 16 keys there are 2 other ones: END key, with a yellow lens, generates a "1" level in the eighth data line (DA07) and the SHIFT key with an orange lens, marked with an arrow (↑), generates a "1" level on the seventh data line (DA06). The illumination of those two push-buttons is not computer controlled. The SHIFT key is illuminated only while being pressed, and the END key is illuminated upon pressing, and stays illuminated until another key (other than SHIFT) is pressed.

Fig. 2.2 REAR VIEW OF THE CABINET

Every press on such a key (other than SHIFT) generates also an interrupt call to the computer, the subroutine of which tells it to read the depressed key's code. The SHIFT key by itself does not generate any interrupt, unless it is pressed in conjunction with another key, the way it is done in conventional keyboards. Note that when the SHIFT key is to be used, it has to be pressed prior to the other key, and kept pressed while the other key is pressed. In this way a completed correct bit combination is guaranteed for the data lines prior to generation of the interrupt call. The unit also generates a power status level, that indicates the presence of power in the unit. The absence of this level indicates that the unit is not available for the computer, whether it is since power is off, or the unit is disconnected from the interface card.

2.2 Electrical Specifications

2.2.1 Joysticks

2.2.1.1 Power requirements:

Voltage:	± 28VDC
Voltage Stability:	± .01%
Current Consumption:	60 mA max (each power supply)

2.2.1.2 Output Characteristics:

Max Voltage Swing:	± 5VDC
Equivalent Output Impedance:	3 KΩ max (each channel)
Minimum Permissible Load Impedance for Error <.01%:	15 MΩ (each channel)
Center Position Output:	OV

2.2.2 Potentiometers:

2.2.2.1 Power Requirements:

Voltage: ± 28 VDC

Voltage stability: ± .01%

Current Consumption: 10 mA max (each power supply)

2.2.2.2 Output Characteristics:

Max Voltage Swing: ± 5VDC

Equivalent Output impedance: 10 KΩ max (each channel)

Minimum Permissible load

Impedance for error <.01%: 40 MΩ (each channel)

Pot. Center position Output: OV

2.2.3 Keyboard

2.2.3.1 Output Voltage level:

High - 4.5 VDC min.

Low - 0.7 max. VDC

2.2.3.2 Output Current level:

High source - 2 mA

Low sink - 200 mA

2.2.3.3 Power requirements:

Voltage: 5 ± .25 VDC

Current: 1.5A max.

2.2.3.4 Power status level: High when OK

Voltage: 5 ± 1VDC

Impedance: 2.2 KΩ max.

2.2.3.5 Data bit level: High when data 0

(output connector) Low when data 1

3.0 DC CIRCUITRY

3.1 Joystick Circuitry

The output of the joystick potentiometers should be within the limit of ± 5V, to comply with the A/D converter requirements. Since the total lever excursion covers only ± 11% of the pot. resistance, a much higher supply voltage has to be applied. For this reason power supplies of ± 28V are employed. The series small resistors are in use for alignment procedure, while output symmetry is accomplished by the joystick trim-tabs.

3.2 Potentiometers Circuitry:

The manual data potentiometers, not like the joysticks, cover their whole resistance range by manual rotation. Therefore, in order to have their output voltage kept within the limit of ± 5V, according to the A/D converter requirements, they have in series much higher resistances. Two separate trimming resistors are also required, since there are no trim=tabs and output voltage centering cannot be achieved by any other way.

3.3 Polarity:

Joystick horizontal output voltage increases towards + 5V value by moving the lever from the left to the right. Vertical output increases similarly when lever is moved down. Manual data potentiometers output voltage increases when pots are rotated clockwise.

3.4 Output Wiring:

Due to the length of the cable, connecting the unit to the A/D card, it is recommended to use true differential input connection, though all input voltages are referred to ground. The common point of the two series 28V power supplies will be used as ground for this case, and will be connected to one of each conductor twisted pairs leaving the unit via the DC output connector.

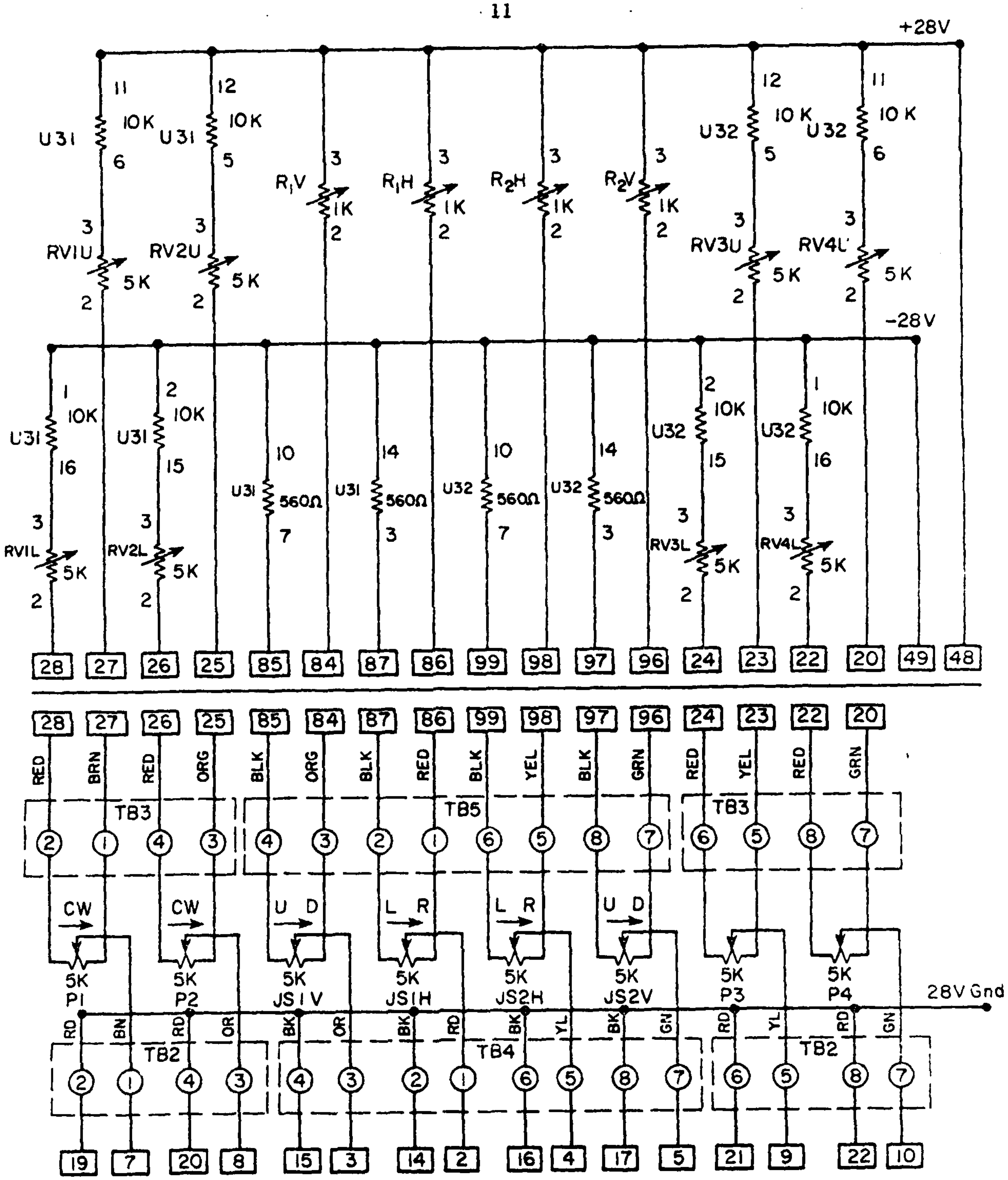


Fig. 3.1.
DC CIRCUIT SCHEMATIC
JOYSTICK CABINET

J1/24 Pins

Hi Row			Low Row		
Pin #	Color	Sig.	Pin #	Color	Sig.
1	Brn	+28V	13	Blk	+28V
2	Red	JS1H+	14	Blk	JS1HGnd
3	Org	JS1V+	15	Blk	JS1VGnd
4	Yel	JS2H+	16	Blk	JS2HGnd
5	Grn	JS2V+	17	Blk	JS2VGnd
6	Blue	28VGnd	18	Blk	28VGnd
7	Brn	P1+	19	Red	P1Gnd
8	Org	P2+	20	Red	P2Gnd
9	Yel	P3+	21	Red	P3Gnd
10	Grn	P4+	22	Red	P4Gnd
11	(Blue)	N.C.	23	(Red)	N.C.
12	Blk	-28V	24	Wht	-28V

All wires twisted pairs.

Table 3.1. DC Output Connector Wiring List.

4.0 LOGIC CIRCUITRY

4.1 Purpose:

Logic circuitry is the term assigned to the circuitry encountered with the keyboard push-buttons. Those keys are mounted in the unit in order to enable keying-in of messages for system control purposes.

4.2 Selected Push-Button Type:

The 16-bits push-button were chosen to be of type SPDT, that is, one normally-closed and one normally-open contacts, with a common point. This was done in order to enable the design of a circuit that will inhibit generation of more than one call by the same press of a button due to contact bouncing. Only after the key returns to its normally closed position, a new call by another press is enabled.

The keys for the END and SHIFT purposes are DPST, due to the other mode they operate. In both cases, the lamp contacts are separate and the light is controlled by external circuitry.

4.3 Electronic circuit:

4.3.1 Encoders:

Since there are no encoders available, that convert one-out-of-16 input into a 4-bit combination, two encoders for one-out-of-eight conversion into 3-bits combination are in use. Thus, the encoded combination occupy 6 bits rather than 4, but there are still unassigned bits in the data word for use.

Since both A and J push-buttons encode an all-zero combination in the encoders, the program might not distinguish between those two buttons. Therefore, the 22_8 and 11_8 encoded combinations were implemented, directly by the line drivers, rather than the encoders (see paragraph 4.3.3).

4.3.2 One-Shot Circuits:

Two sequential one-shot multivibrators are operated by every press of a key (again, excluding SHIFT). The first, of 500μSec, yields a delay to enable skipping all the bouncing transients that might occur when a key is pressed. The second pulse, of 100μSec, is used as a strobe for loading the input data register and setting the DONE flip-flop, which asserts in turn the interrupt call. The latter mentioned electronic circuits, as the input data register and the DONE flip-flop are not located in the unit but on the interface card.

4.3.3 Line Drivers:

The logic outputs to the back-panel connector are buffered by line-drivers type LM75451P. END lamp is also driven by such a driver. Only the SHIFT and PWL (power status bit) which are not generated by TTL circuits but directly by the push-button contacts or the power-line, are connected directly to the connector. Every driver has two inputs AND functioned, and the unused inputs are generally connected to the Hi level, except for the two cases, that the combinations 11_8 and 22_8 have to be encoded. This is done directly through the other inputs, that are not connected to the encoder outputs.

4.4 Status Bits:

The only status bit generated within the unit is the Power status (PWL) which appears in the 15th level (DA14). The other 2 bits, ERROR (DA15) and DONE (DA13), are generated in the interface card. The check level is generated by the +5V supply power source, in series with a 2200Ω resistance.

4.5 Data Input Word Format:

The final structure of the input word will be as follows: The Low Order Byte (L.O.B.) shows data information (pressed push-buttons). The High Order Byte (H.O.B.) shows the status by the three most significant bits. The other bits are not applicable.

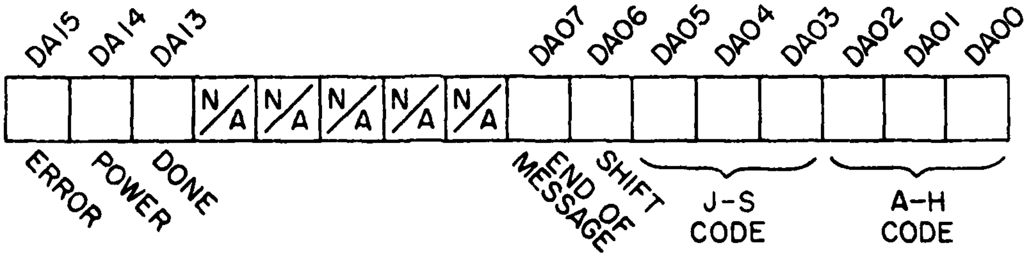


Fig. 4.1. Input Data Register Format

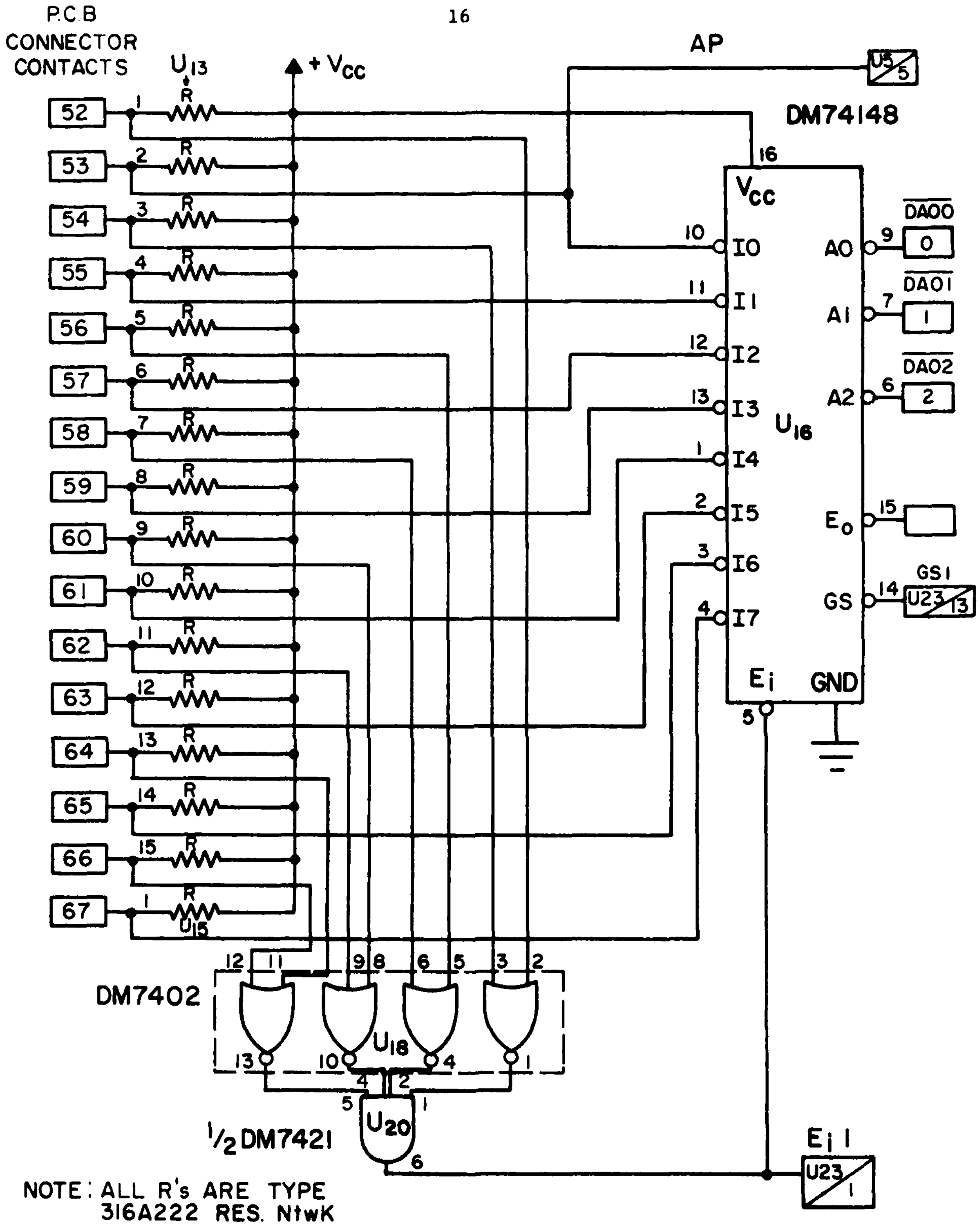
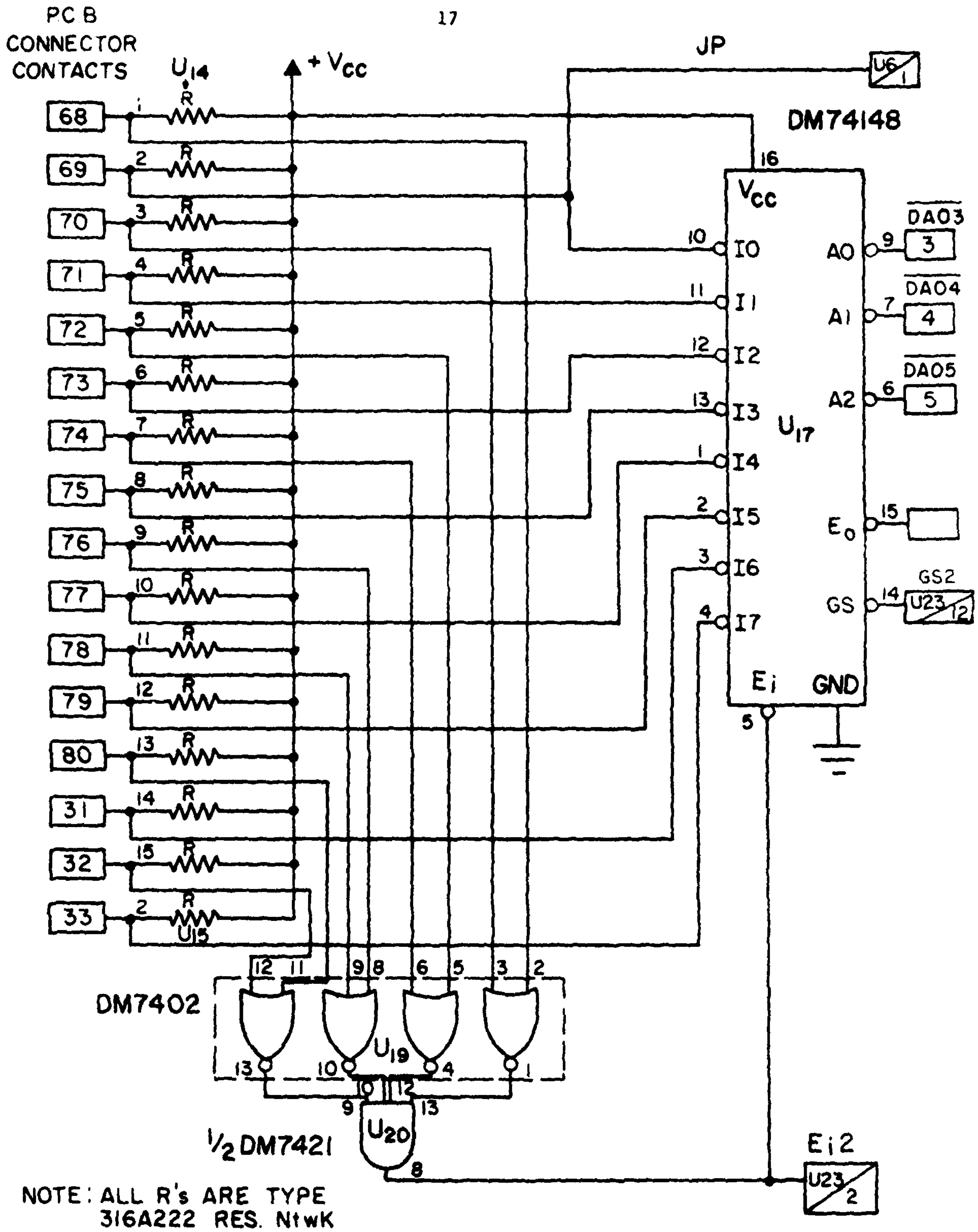


Fig. 4.2.a.
JOYSTICKS & KEYBOARD CABINET
LOGIC SECTION/ENCODING CIRCUIT

LEAST SIGNIFICANT



NOTE: ALL R's ARE TYPE
316A222 RES. NtwK

Fig. 4.2.b.
JOYSTICKS & KEYBOARD CABINET
LOGIC SECTION/ENCODING CIRCUIT

MOST SIGNIFICANT

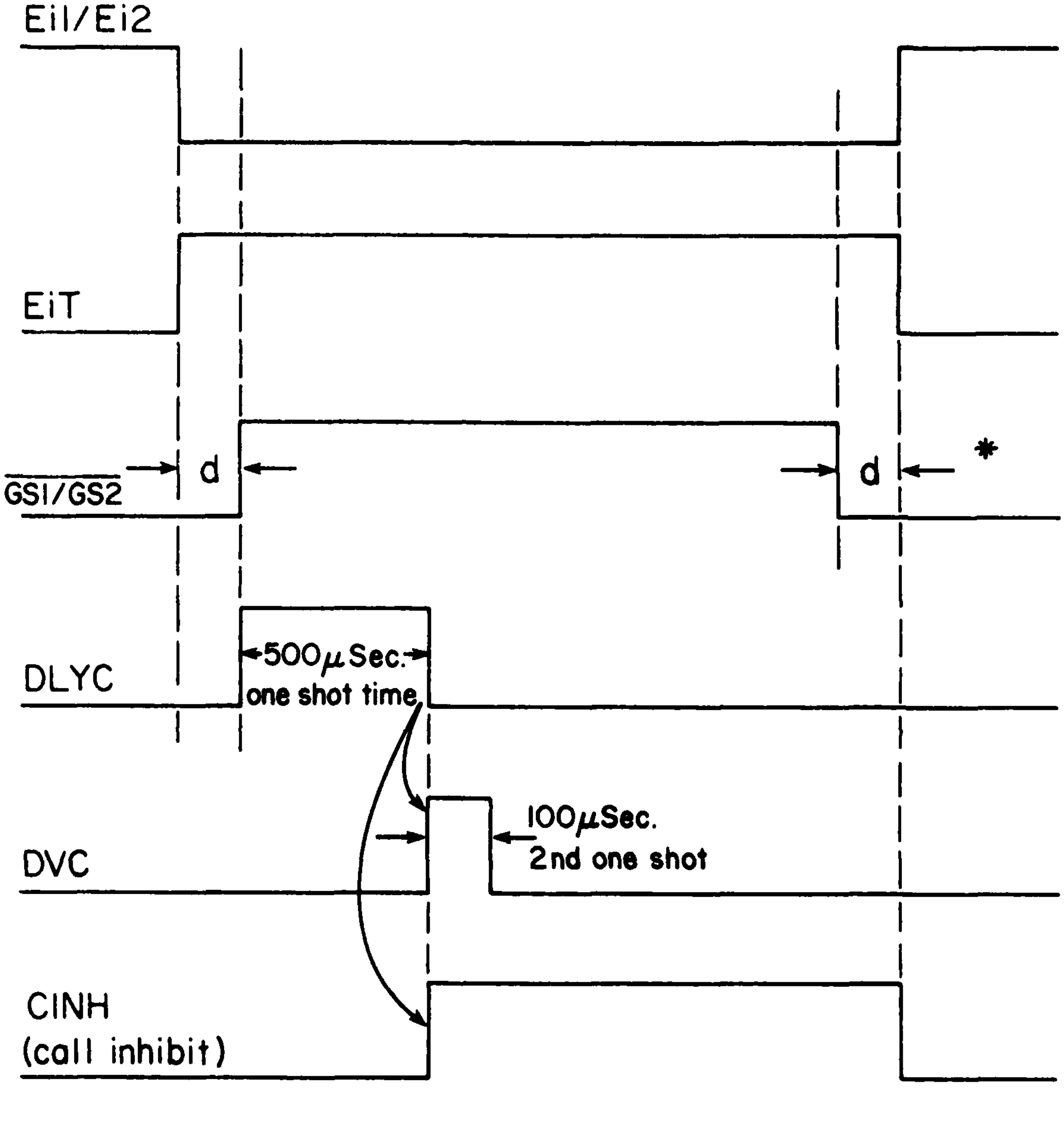


Fig. 4.3.
LOGIC SECTION TIMING DIAGRAM

*d - The delay implied by the distance between breaking the N.C. contacts and making the N.O. contacts

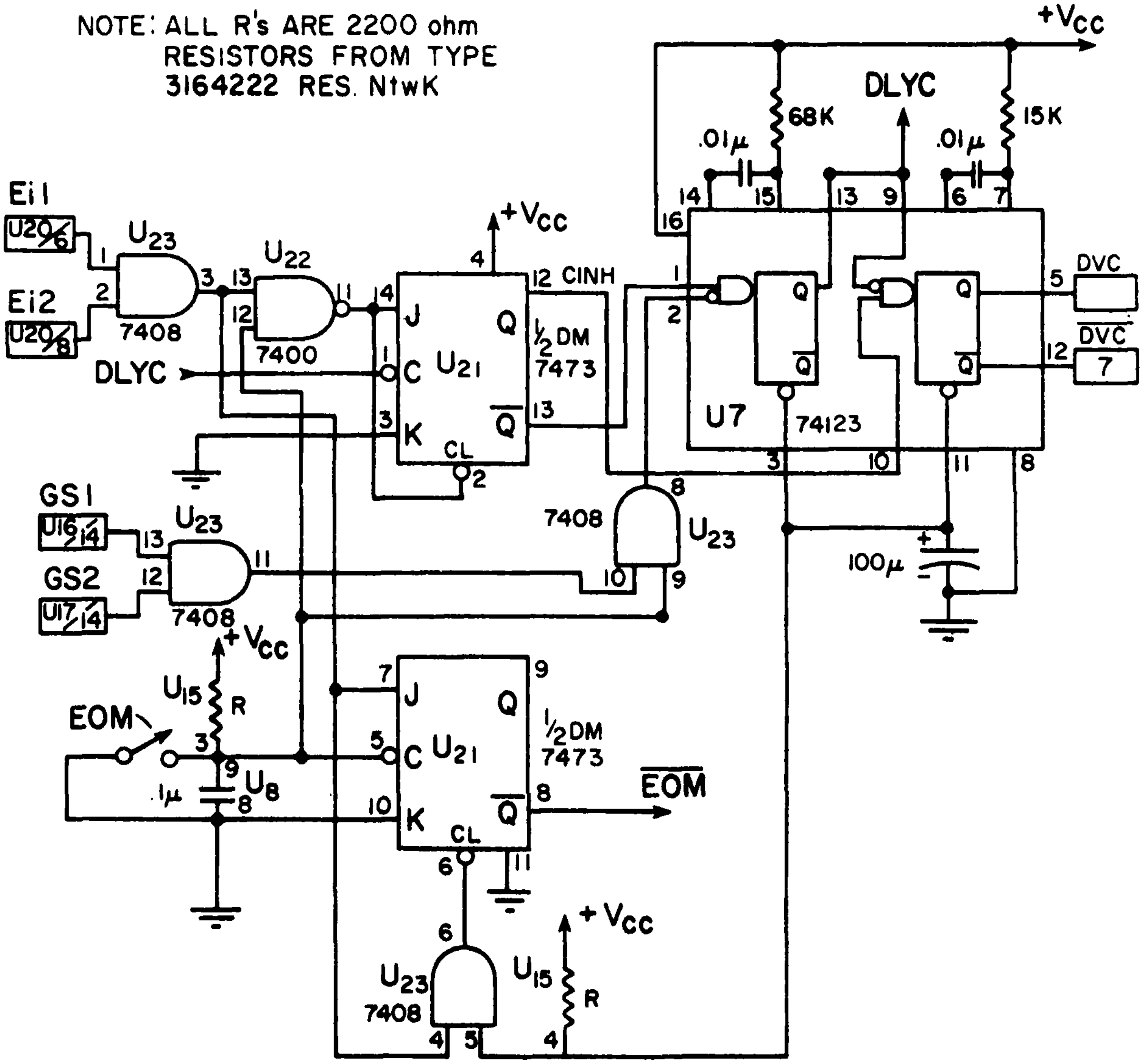
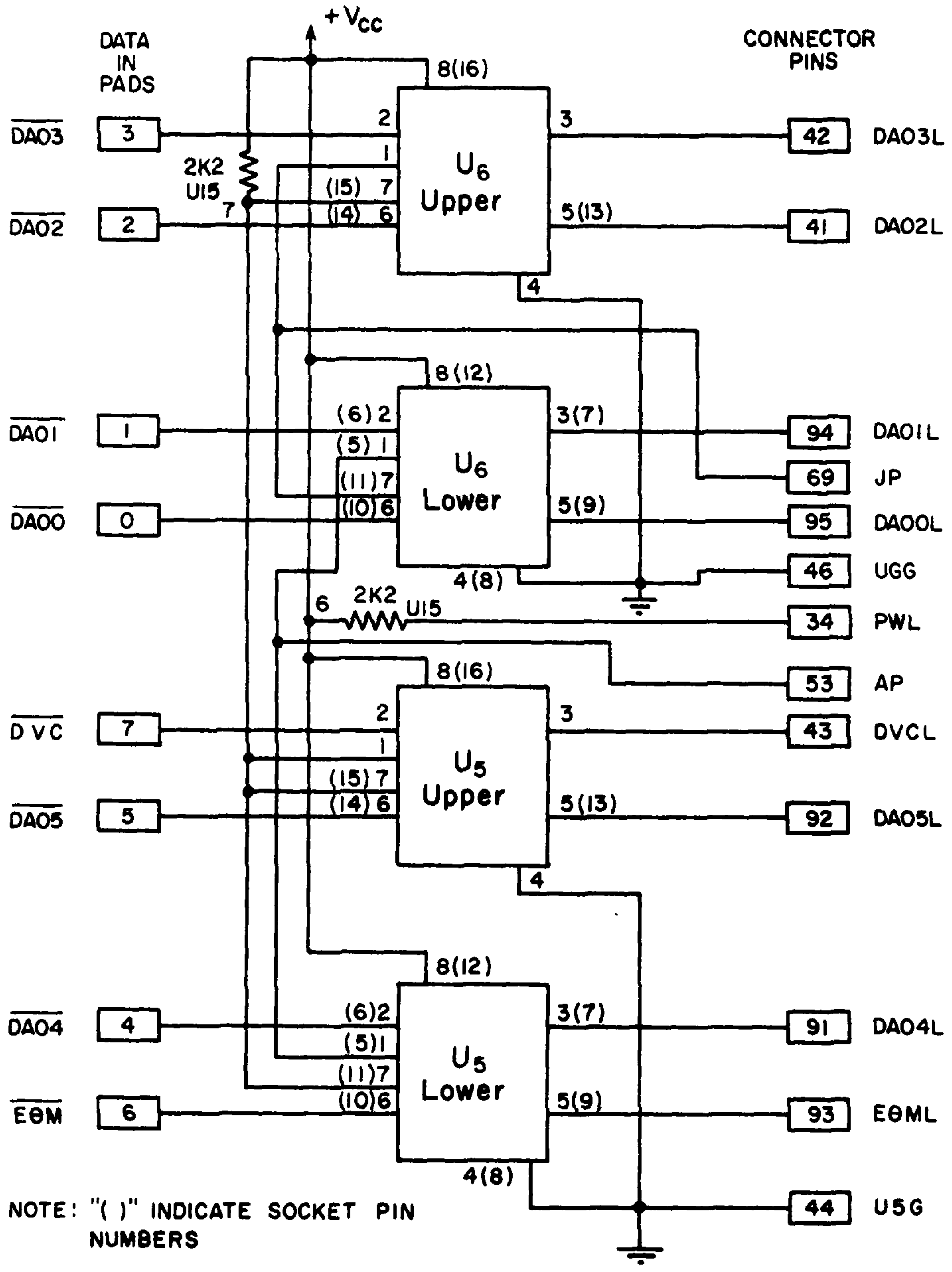
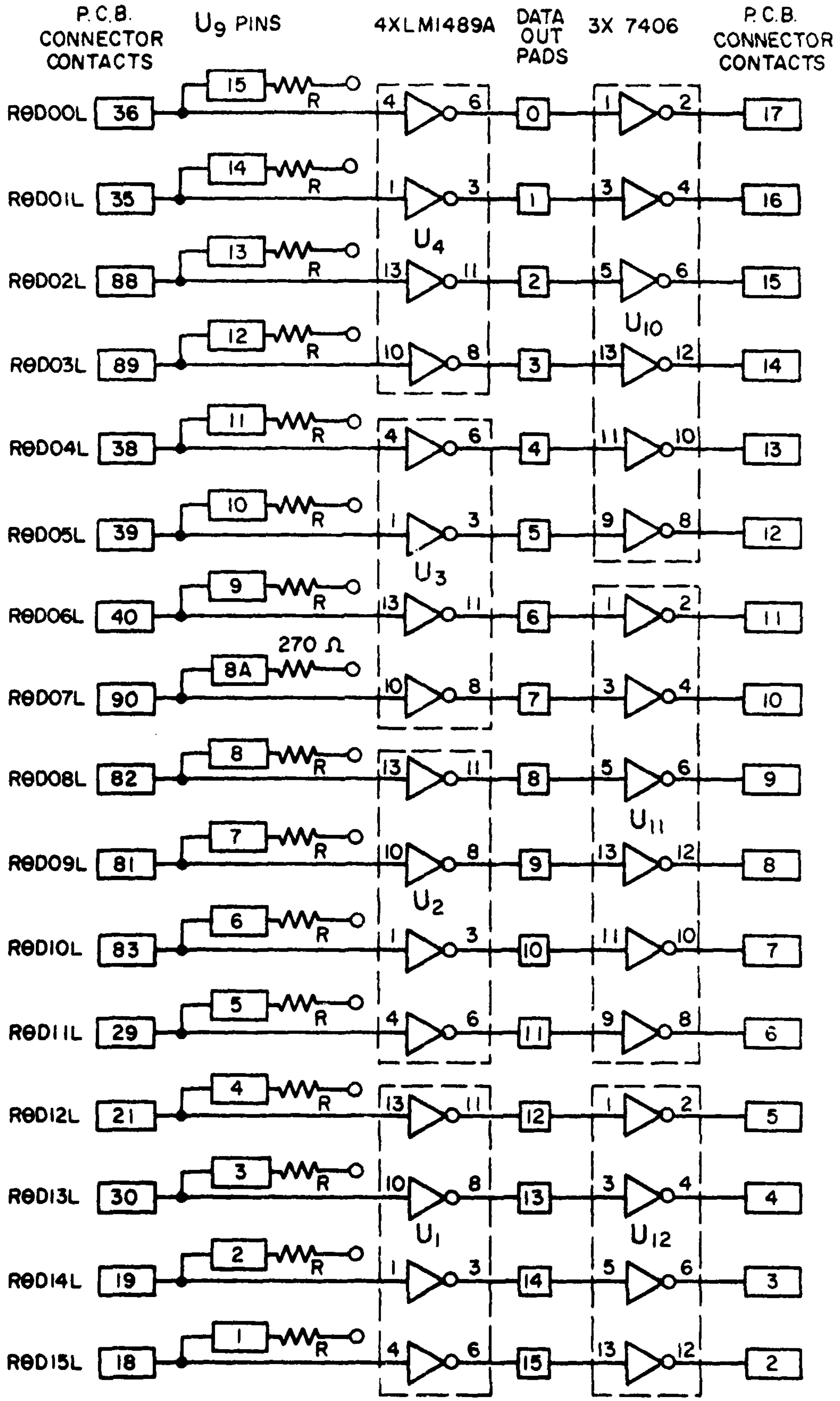


Fig. 4.4.
JOYSTICKS & KEYBOARD CABINET
LOGIC SECTION/SHIFT, EOM & DEVICE CALL CIRCUITS



NOTE: "( )" INDICATE SOCKET PIN NUMBERS

Fig. 4.5.
JOYSTICK & KEYBOARD CABINET
LOGIC SECTION/OUTPUT SIGNALS


P.C.B. CONNECTOR CONTACTS
U9 PINS
4XLM1489A
DATA OUT PADS
3X 7406
P.C.B. CONNECTOR CONTACTS
RΘD00L 36 15 R 4 6 0 1 2 17
RΘD01L 35 14 R 1 3 1 3 4 16
U4
RΘD02L 88 13 R 13 11 2 5 6 15
U10
RΘD03L 89 12 R 10 8 3 13 12 14
RΘD04L 38 11 R 4 6 4 11 10 13
RΘD05L 39 10 R 1 3 5 9 8 12
U3
RΘD06L 40 9 R 13 11 6 1 2 11
270 Ω
RΘD07L 90 8A 10 8 7 3 4 10
RΘD08L 82 8 R 13 11 8 5 6 9
U11
RΘD09L 81 7 R 10 8 9 13 12 8
U2
RΘD10L 83 6 R 1 3 10 11 10 7
RΘD11L 29 5 R 4 6 11 9 8 6
RΘD12L 21 4 R 13 11 12 1 2 5
RΘD13L 30 3 R 10 8 13 3 4 4
U1
U12
RΘD14L 19 2 R 1 3 14 5 6 3
RΘD15L 18 1 R 4 6 15 13 12 2

NOTE: ALL R RESISTORS ARE LOCATED IN A 316A222 RESISTOR NETWORK
THEIR OTHER END IS CONNECTED TO +5V.

Fig. 4.6. JOYSTICK & KEYBOARD CABINET LOGIC SECTION/LAMP DRIVING CIRCUIT

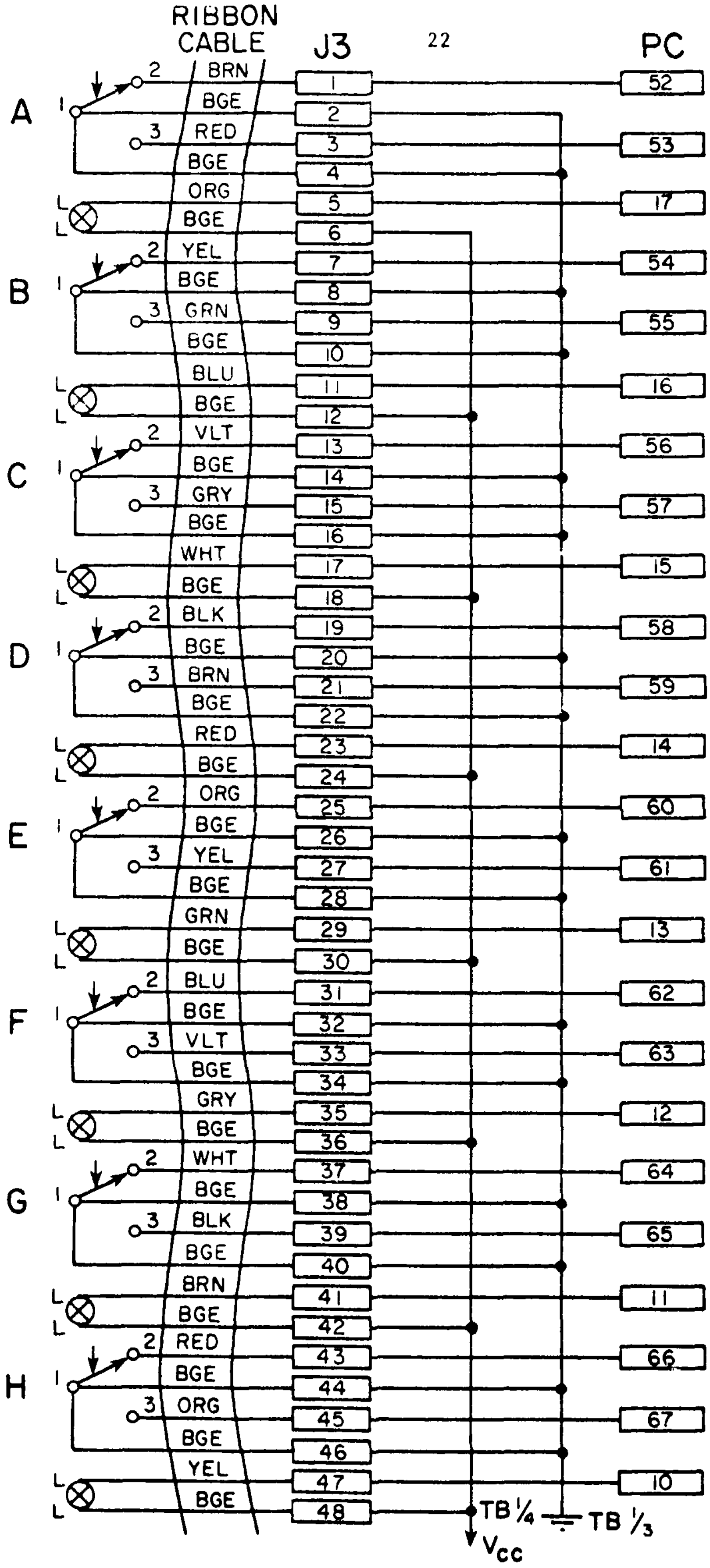


Fig. 4.7.a.
LAMP CIRCUIT LEAST SIGNIFICANT

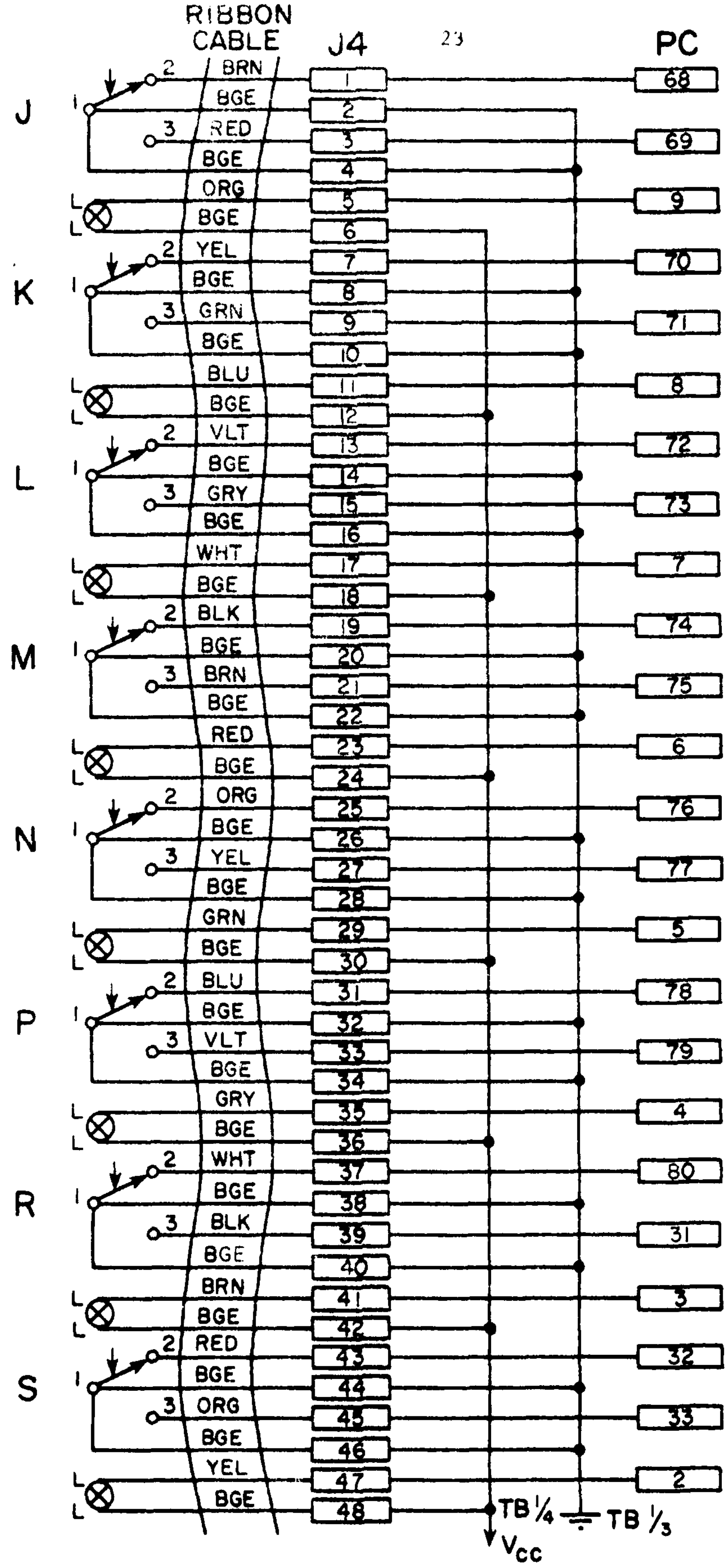


Fig. 4.7.b.
LAMP CIRCUIT MOST SIGNIFICANT

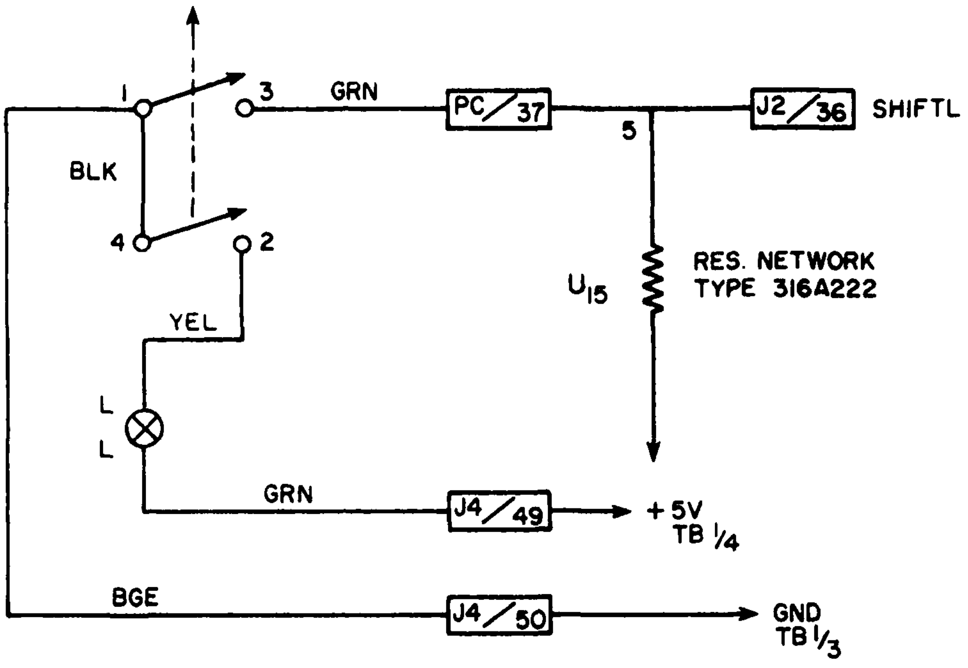


Fig. 4.7.c.
SHIFT CIRCUIT

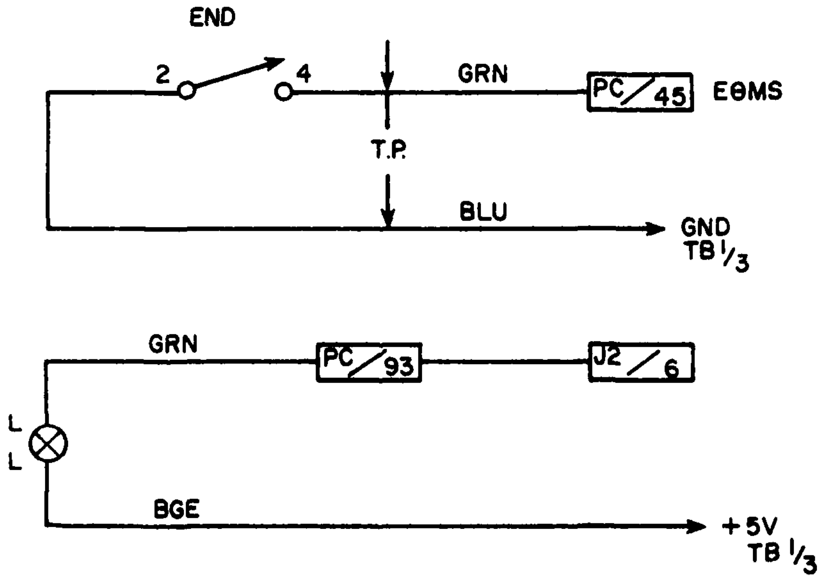


Fig. 4.7.d.
EOM CIRCUIT

CONNECTOR WIRING LIST

J2/36 Pins

Hi Row			Lo Row		
Pin #	Color	Sig.	Pin #	Color	Sig.
1	Brn	+5V	19	Blk	5VGnd
2	Red	A(RDθ00L)	20	Blk	E(RDθ04L)
3	Org	B(RDθ01L)	21	Blk	F(RDθ05L)
4	Yel	C(RDθ02L)	22	Blk	G(RDθ06L)
5	Grn	D(RDθ03L)	23	Blk	H(RDθ07L)
6	Blu	EθML	24	Blk	EθμGnd
7	Brn	DA05	25	Red	DA05Gnd
8	Org	DA04	26	Red	DA04Gnd
9	Yel	DA03	27	Red	DA03Gnd
10	Grn	DA-2	28	Red	DA02Gnd
11	Blu	DA01	29	Red	DA01Gnd
12	Blk	DA00	30	Wht	DA00Gnd
13	Red	DVCL	31	Wht	DVCLGnd
14	Grn	J(RDθ08L)	32	Wht	N(RDθ12L)
15	Blu	K(RDθ09L)	33	Wht	P(RDθ13L)
16	Brn	L(RDθ10L)	34	Grn	R(RDθ14L)
17	Org	M(RDθ11L)	35	Grn	S(RDθ15L)
18	Yel	PWL	36	Grn	SHIFTL

←--------- all wires twisted pairs ---------→

Table 4.1 Logic Output Connector J2 Wiring List.

5.0 POWER SUPPLIES

5.1 Requirements:

General requirements from the power supplies are as follows:

1) 2 28V stabilized sources are required for the DC System. The sources have to be connected in series to form a dual-symmetry positive-negative source.

2) 1 5V source for the logic supply (including bulbs).

3) The current requirements are as shown in the specification table (p. 2.2), i.e. each 28V supply should yield at least 70 mA, and the 5V supply should yield at least 1.5A.

5.2 Power Supply Chosen:

Practically, type 22 SM series submodular Powertec power units are employed. Two units are of type 22A-500, that are used as 28V 0.475A sources. The unit used as 5V power supply is of type 22B-100, that can yield 5V at 3A current.

5.3 Fuses:

Each power supply has its own fuse, over voltage protector and current limiter. Therefore, no additional fuses or other protecting devices are employed in the unit.

5.4 Power Supply Data:

Detailed data sheets of the submodular power units are to be found in Appendix B.

5.5 Transformers:

Two separate transformers are used for power supply. Stancor type p-8653 is used for 5V supply, and Stancor type p-8675 (with dual secondary winding) is used for the 2 28V supplies.

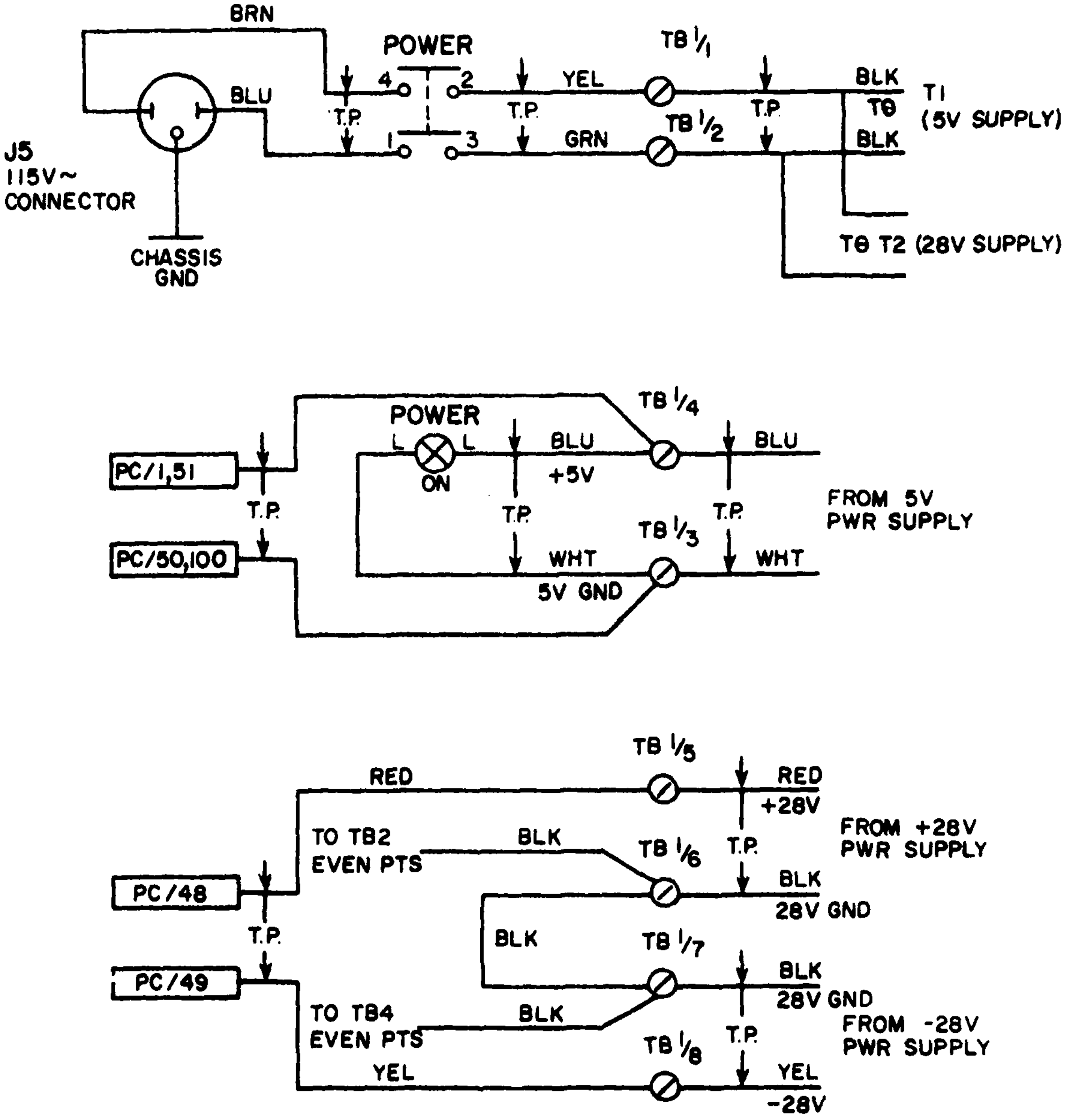


Fig. 5.1.
TB1-POWER SUPPLY TERMINAL BOARD CONNECTIONS

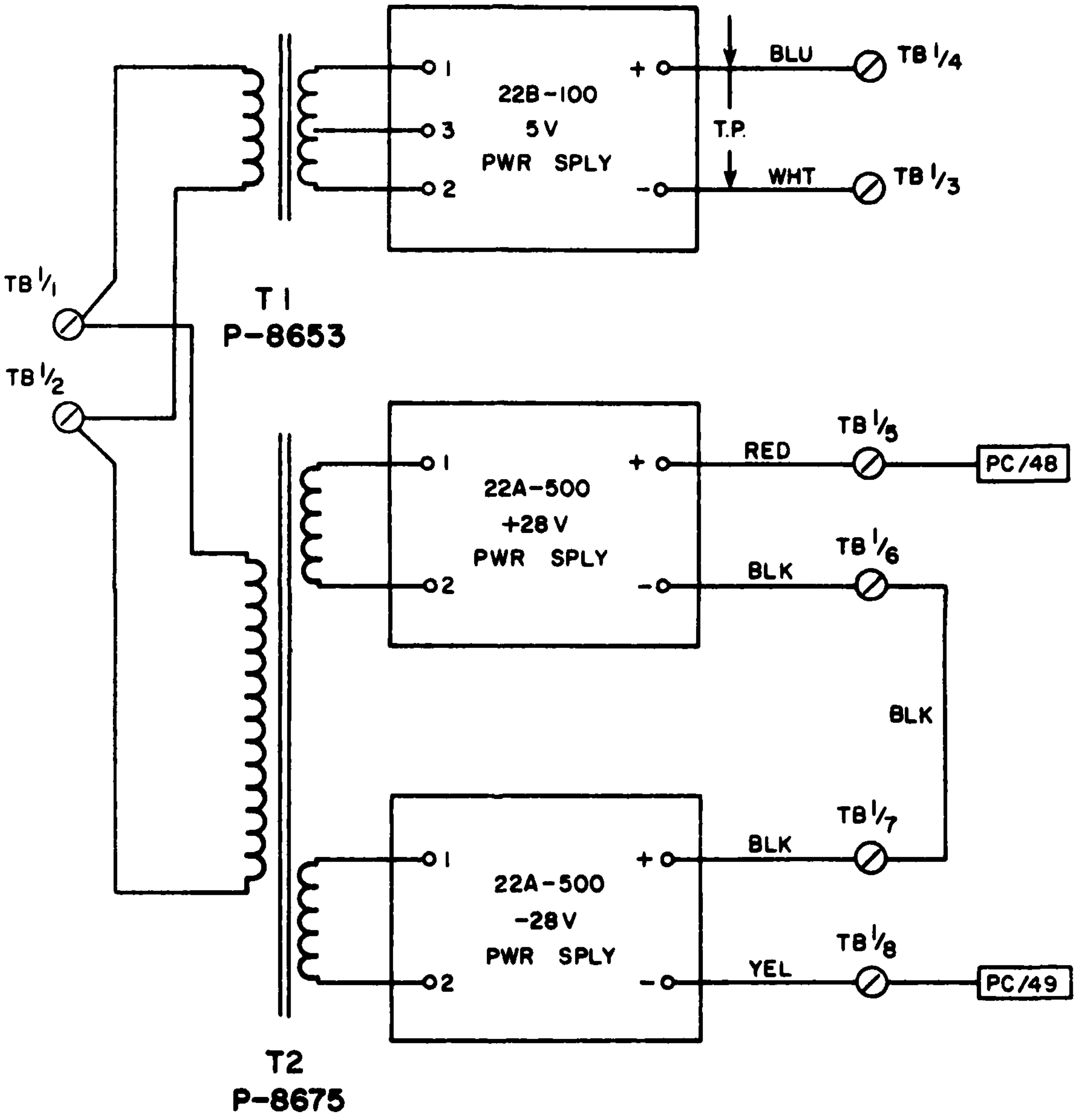


Fig. 5.2.
DC POWER SUPPLY CONNECTIONS

6.0 TEST AND ALIGNMENT PROCEDURE

6.1 Power Supply Voltage Adjustment:

6.1.1 Connections and Turning the Unit On:

a) Connect unit to Test-Box by plugging CC1 to receptacle J1 in the Joystick Cabinet rear panel, and CC2 to the receptacle J2. Also connect a DC voltmeter or DVM to the top binding posts. Test-Box is described in Appendix C.

b) Plug Power Cord to 115VAC power connector.

c) Turn the unit on by pressing POWER push-button. Note that green lens of push-button is illuminated, as well as PWR ST indicator on the Test-Box. No other indicator light on unit or on Test-Box should be illuminated.

6.1.2 Adjustments:

6.1.2.1 +28V adjustment:

a) Set selector-switch on front panel of Test-Box to +28V position.

b) Notice reading of voltmeter. Reading should be 28 ± 0.1V. If reading is different, adjust +28V power supply using a long, thin screwdriver.

6.1.2.2 -28V adjustment:

a) Set selector-switch on Test-Box to -28V position.

b) Adjust reading to 28V as in 6.1.2.1b. No need to invert polarity of voltmeter probes.

6.1.2.3 +5V adjustment:

a) Set selector-switch on Test-Box to 5V position.

b) Adjust voltage to 5 ± 0.1V and, if necessary, select a better measurement range of voltmeter.

6.2 DC Section Test:

6.2.1. Connections and Presetting:

a) Connect a 10VDC voltmeter to the top binding-posts of the Test-Box. The voltmeter should be preferably a ± 10V reading with a zero-centered pointer, or a ± reading DVM. Positive input to red, negative input to black binding posts.

6.2.2. Joystick Output Adjustments:

a) Move J.S. #1 (Left-hand) to its extreme right position. Vertical position needs not to be changed.

b) Note that VM reading should be +5V. If it is not, use trimpot R1H for adjustment.

c) Move J.S. #1 to extreme left position. The VM reading should now be -5V. If it is not, use horizontal trim-tab for adjustment.

d) Repeat steps a, b and c for several times until output voltages are ±5V at the joystick extreme positions. Output at the center position should be zero.

e) Turn rotary selector switch to next position, JS1V

f) Move J.S. #1 lever to extreme lower position and adjust reading of VM to +5V, using trimpot R1V.

g) Move J.S. #1 lever to extreme upper position, and adjust reading of VM to -5V, using vertical trim-tab for adjustment.

h) Repeat steps f and g for several times, until output voltages are ± 5V at the joystick extreme vertical positions. No need of changing horizontal lever position during vertical alignment.

i) Repeat steps a through h for J.S. #2 (right-hand), using selector switch position JS2H and JS2V in lieu of JS1H and JS1V, respectively. Adjust output voltages using trimpots R2H and R2V in lieu of R1H and R2V, respectively. Negative outputs should be adjusted using J.S. #2 trim-tabs.

j) Note that after alignment procedure, all 4 joystick output readings should be zero, if both joysticks are in center position.

k) Lock joystick levers using locking screws, and check that they are locked in position.

6.2.3. Manual Data Potentiometers Alignment:

a) Turn rotary selector-switch on Test-Box to position P1.

b) Turn Pot #1 to maximum clockwise (CW) position, and adjust VM reading to be +5V, using RV1U trimpot for adjustment.

c) Turn Pot. #1 to maximum counter-clockwise (CCW) position, and adjust VM reading to be -5V, using RV1L trimpot for adjustment.

d) Repeat steps b and c for several times, until no more adjustments are necessary for Pot. #1.

e) Turn selector switch to position P2, and repeat procedure described in b, c, and d for Pot. #2, using RV2U and RV2L in lieu of RV1U and RV1L, respectively.

f) Use the same procedure for adjustments of Pots. #3 and #4, setting selector switch to positions P3 and P4, respectively. For adjustments, use RV3U and RV3L for P3, and RV4U and RV4L for P4.

6.3 Logic Section Test:

6.3.1. Auxiliary Keys Test:

a) Press SHIFT push-button and note that all indicating LED's on the Test Box panel are dark, except PW ST, which will stay illuminated through all the procedure.

b) Press END push-button and note that EOM indicator will be illuminated, in conjunction with DVC indicator on the Test-Box panel. The push-button itself should be also illuminated. Both indicators should stay illuminated after push-buttons are released.

c) Press SHIFT and note that DVC indicator will turn off. The push-button itself should be lighted only when pressed.

d) Note that EOM indicator on the Test-Box panel and END push-button are still illuminated.

6.3.2. 16 Main Keys Test:

a) Press the key marked A. Note that the key itself is not illuminated.

b) Note that EOM indicator will turn off, and DVC will turn on.

c) Among the indicators DA0 to DA5, the indicators DA1 and DA4 only will be illuminated.

d) Release A key. DA2 and DA4 should turn off. DVC should remain illuminated.

e) Press SHIFT key and note that DVC indicator will turn off. SHIFT push-button itself will be illuminated only when pressed.

f) Repeat steps a through e* for all other 15 keys, B through

* Since EOM test is not repeated, turn-off of EOM as indicated in step b will not repeat for other 15 keys test.

S. Illumination code of DAO through DA5 for all keys is as follows: (N-ON, F-OFF)

Key	DA5	DA4	DA3	DA2	DA1	DAO
A	F	N	F	F	N	F
B	F	F	F	F	F	N
C	F	F	F	F	N	F
D	F	F	F	F	N	N
E	F	F	F	N	F	F
F	F	F	F	N	F	N
G	F	F	F	N	N	F
H	F	F	F	N	N	N
J	F	F	N	F	F	N
K	F	F	N	F	F	F
L	F	N	F	F	F	F
M	F	N	N	F	F	F
N	N	F	F	F	F	F
P	N	F	N	F	F	F
R	N	N	F	F	F	F
S	N	N	N	F	F	F

Table 6.1 Keyboard Code

6.4 Push-Buttons Illumination Test:

a) Note that all white push-buttons illumination lenses are dark.

b) Press Lamp-Test push-button on the Test-Box panel, and note that all push-button illumination lenses turn to be lighted. They will return to the previous position when the test push-button is released.

c) Note that this test concerns only the white lens push-buttons, and not END or SHIFT.

6.5 Disconnecting Unit from Test-Box:

Turn power off, then disconnect CC1 and CC2 from J1 and J2, respectively.

ATTENTION!

The unit has a built-in power supply! Remember to turn unit power off before connecting to interface cards. Power may be restored only after connection to interface is accomplished.

7.0 INTERFACING MODULES

General information:

Since the unit operates as two units without any necessary connection between each other, it is interfaced to the computer by two separate interface units. The DC section is interfaced through an A/D card Model 635/11AD, manufactured by ADAC. the logic section is interfaced through a general purpose interface module type MDB-11C, manufactured by MDB Systems, Inc.

Those two cards are described in Appendices D and E, respectively. However, several specific wiring works had to be done for fitting those cards to this specific application, especially in the MDB card, where user's interface circuits had to be added. This chapter describes those specific circuitries.

7.a. Analog to digital interface.

7.a.1. Card Model:

There are many options that can be purchased together with the card type 635/11AD.

7.a.1.1 Number of Channels:

The smallest number of channels possible on this card is 16, if it is used as a single ended input. But since about 30' of cable connect the DC source (Joystick unit) to the card, differential input mode has to be used, and this reduces the effective number of channels by one half, i.e. it is now 8 channels.

7.a.1.2. Input Voltage Range:

Practical value is ± 5V (see 3.1)

7.a.1.3. DAC's

No DAC (Digital to Analog Converters) are needed now. It is possible to add ones in the future.

7.a.1.4 DMA Compatibility:

No DMA Compatibility is required. Hence, the resulting type number is as follows:

635/11AD-8DI-C-0-0-0. See details in Module instruction manual.

7.b. MDB-11C Interface Card:

7.b.1

The card includes all the required logic for connections to the UNIBUS, and is described in Appendix E. Anyhow, additional user's interface is required, to enable connections to the joystick cabinet. This additional logic is described in this chapter.

7.b.2. Flow Chart:

The flow chart described consists of two parts: the one concerning the interrupt logic on the left, and the one concerning read and write operations on the right. Naturally, the right operation proceeds the left, but the writing operation is by nature independent of the interrupt. Besides, in case of an error, the processor might initiate a read operation without DONE ready, and this is a possibility that has to be considered. Therefore, the reading and writing series of steps is described independently.

7.b.3. Block-Diagram:

The block diagram is divided into 3 parts: the left hand includes the section built in the joystick cabinet, and its description is given in detail in Chapter 4. The right hand part includes the logic already built on the interface card, which is discussed in Appendix E. The mid-part is the user's interface which is described in this chapter, and the schematics of which are given here.

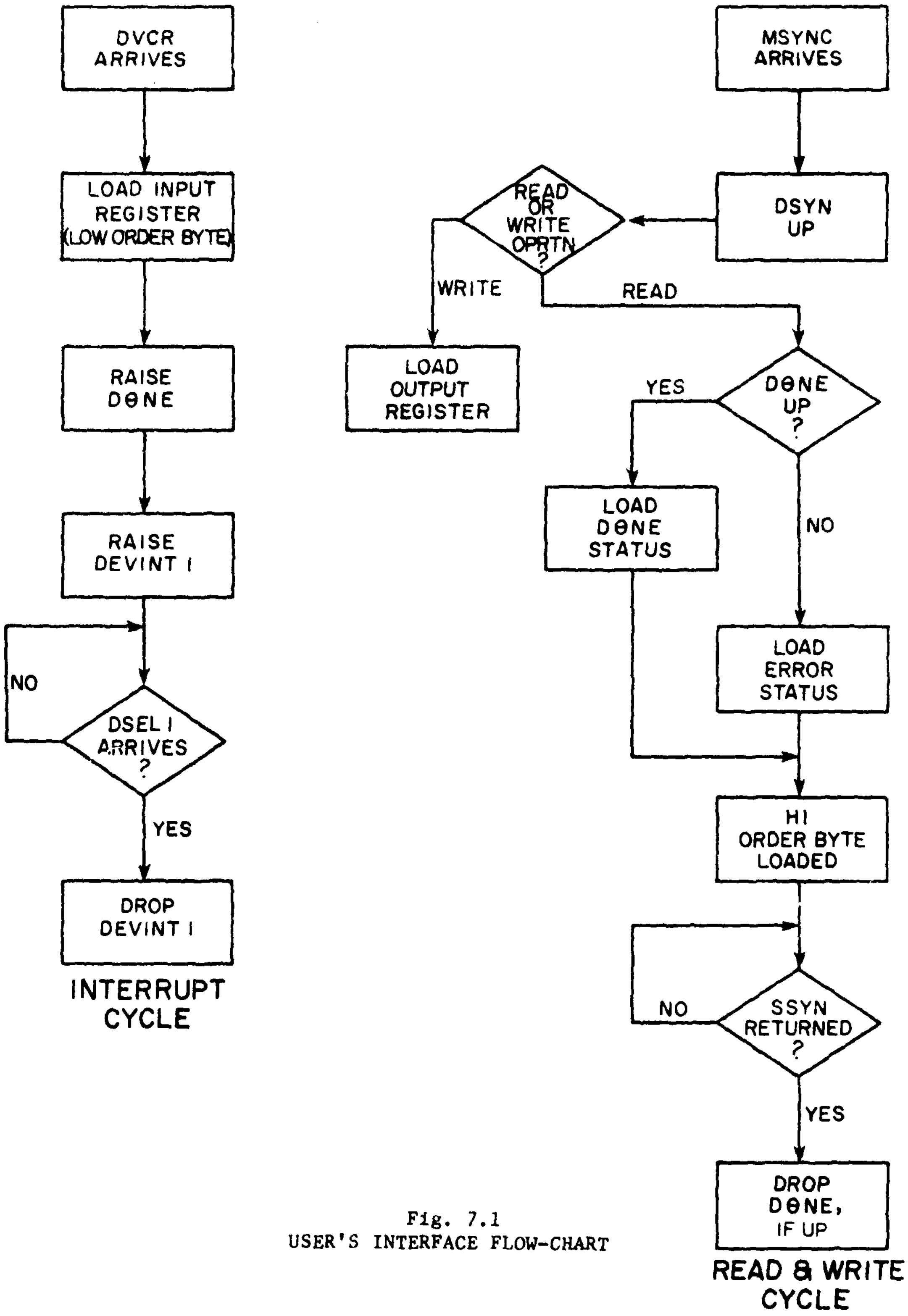
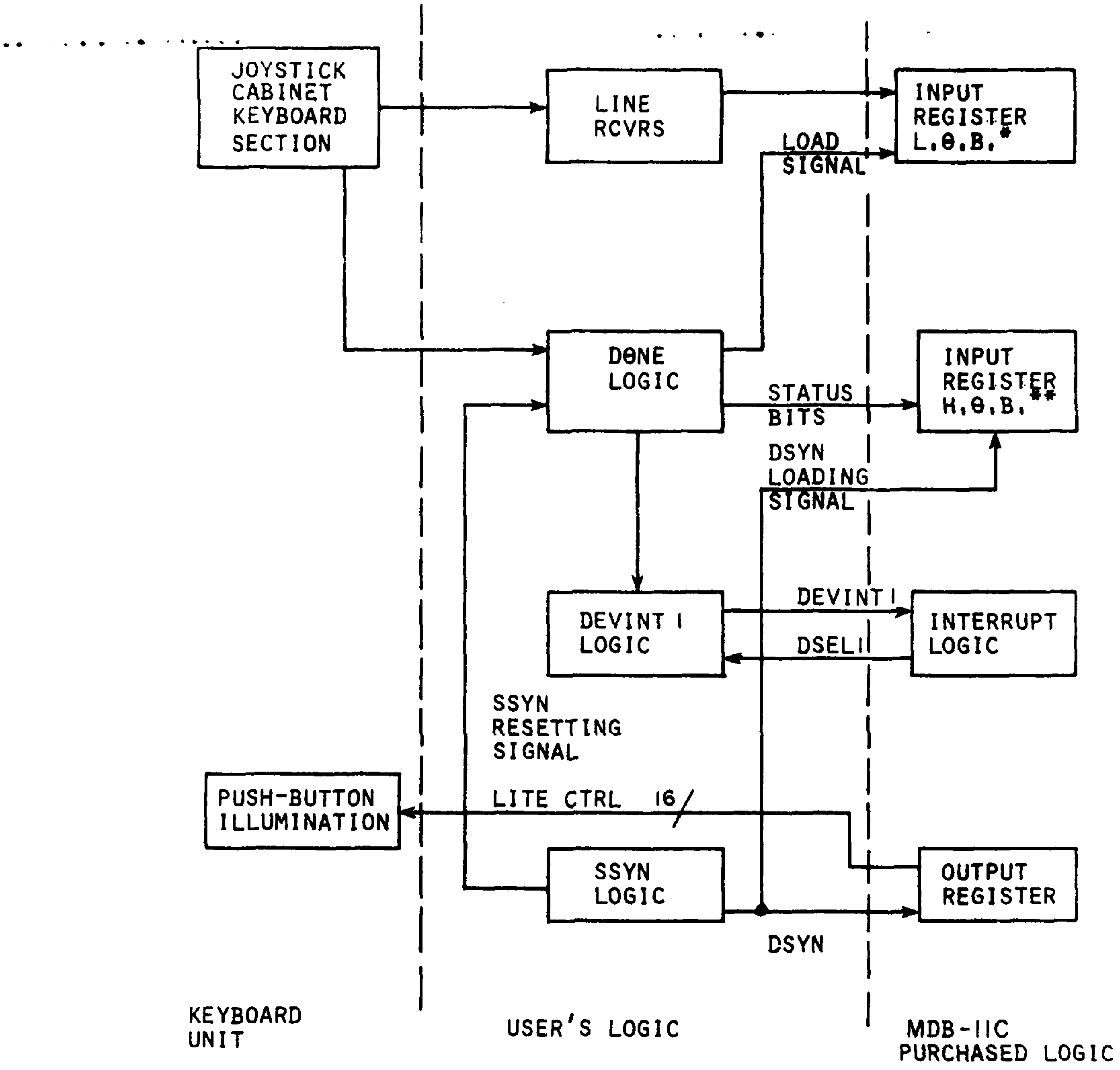


Fig. 7.1
USER'S INTERFACE FLOW-CHART



*L.O.B. - LOW ORDER BYTE

**H.O.B. - HIGH ORDER BYTE

Fig. 7.2.
KEYBOARD INTERFACE
BLOCK DIAGRAM

7.b.4. Logic Schematics:

No signals are transferred from the keyboard circuit to the interface through line receivers: the 6 encoded bits, the auxiliary keys END and SHIFT, and the signals DVCL and PWL (Power status bit). All of them, except SHIFTL and PWL are transferred through twisted-pairs conductors, because of their speed. The other two are DC signals and can be transferred in single leads.

Loading of the input register is divided into two bytes, the Low Order Byte (LOB) that includes the date from the keyboard is loaded by the DVC signal. The H.O.B. which is the status part of the word, and depends on the circumstances the input word is read by the processor, is loaded by the DSYN pulse of a read operation. The lamp control circuit is connected to the output register and operates entirely separated from the other circuitry.

7.b.5. Jumpers:

7.b.5.1. Address decoding:

The address assigned for the keyboard device is 764020. This address location is used both for reading the combined input byte and status byte word, as well as for writing in the output register the illuminated key bulbs combination. However, due to the internal construction of the MDB-11C module, the board will respond also to addresses 764022, 024, and 026, although those locations are not in use. Hence, those locations must not be addressed by any program.

7.b.5.2. Interrupt Vector:

The interrupt vector for the keyboard is assigned to be 324 (11010100). The bus-request process takes place at priority level 4 (BR4-BG4).

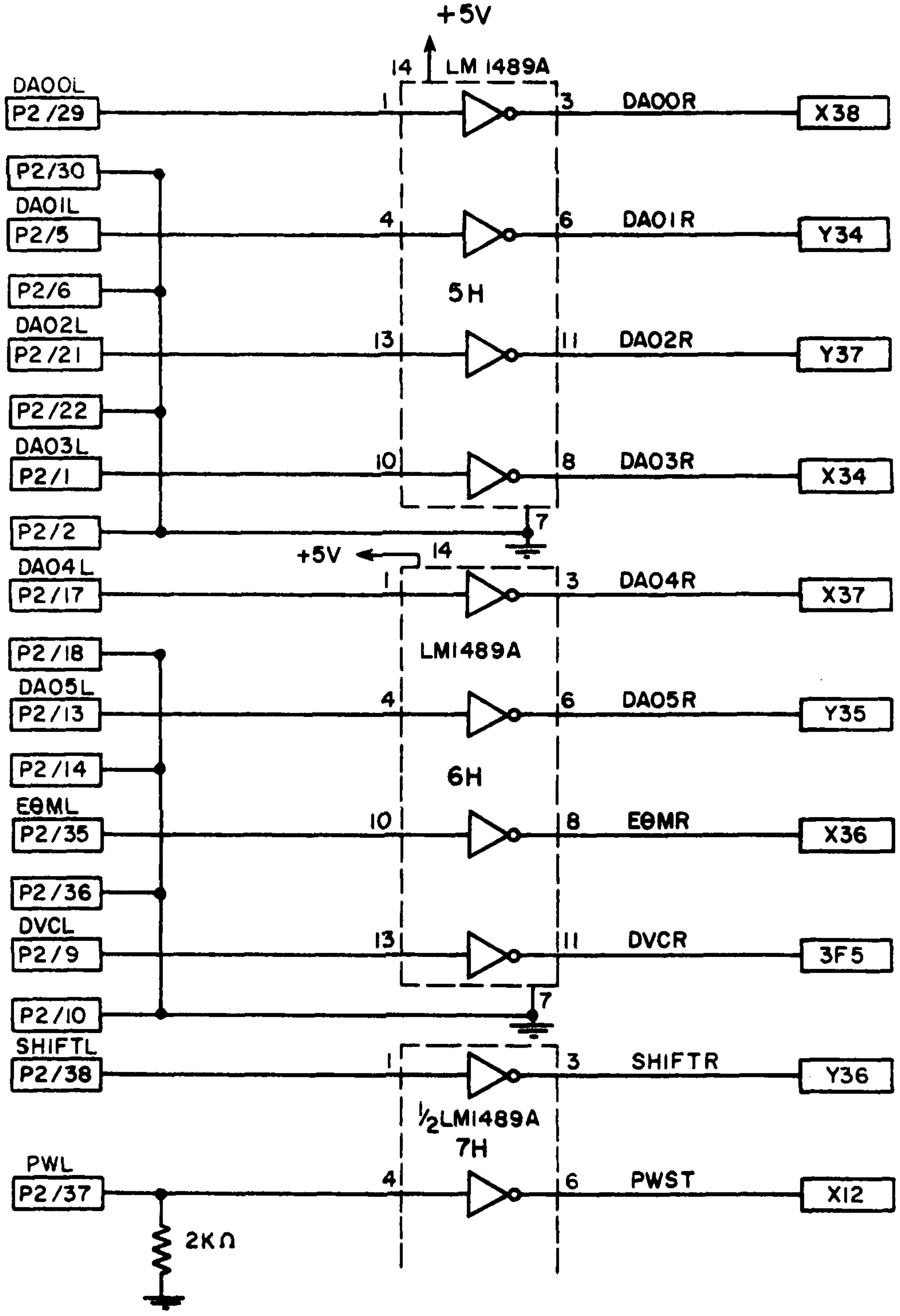


Fig. 7.3.
MDB-11C INTERFACE CARD
LINE RECEIVERS CONNECTIONS

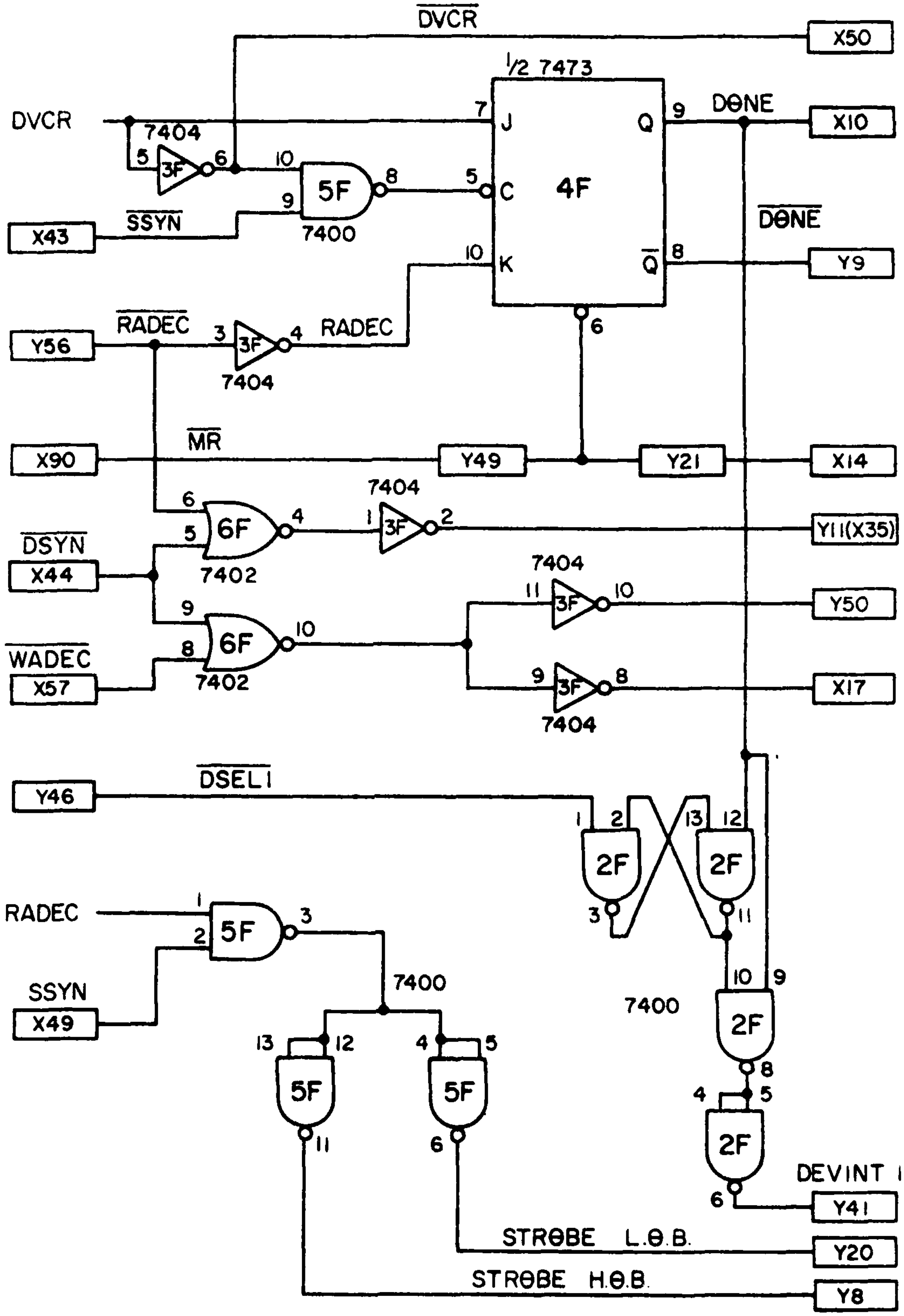


Fig. 7.4.
MDB-11C INTERFACE CARD
REGISTERS CONTROLS & INTERRUPT LOGIC

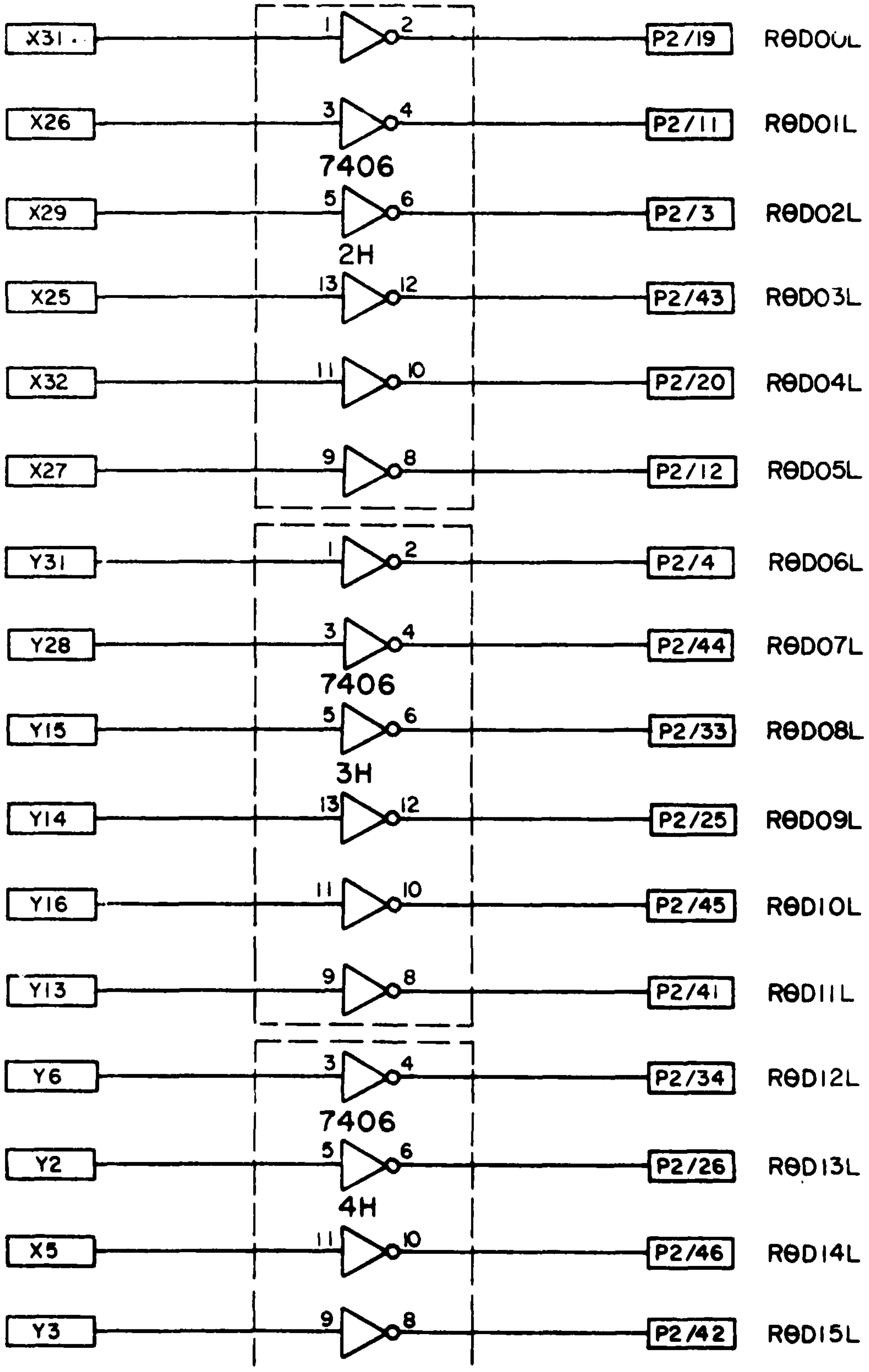


Fig. 7.5.
MDB-11C INTERFACE CARD
LAMP CONTROL REGISTER OUTPUTS

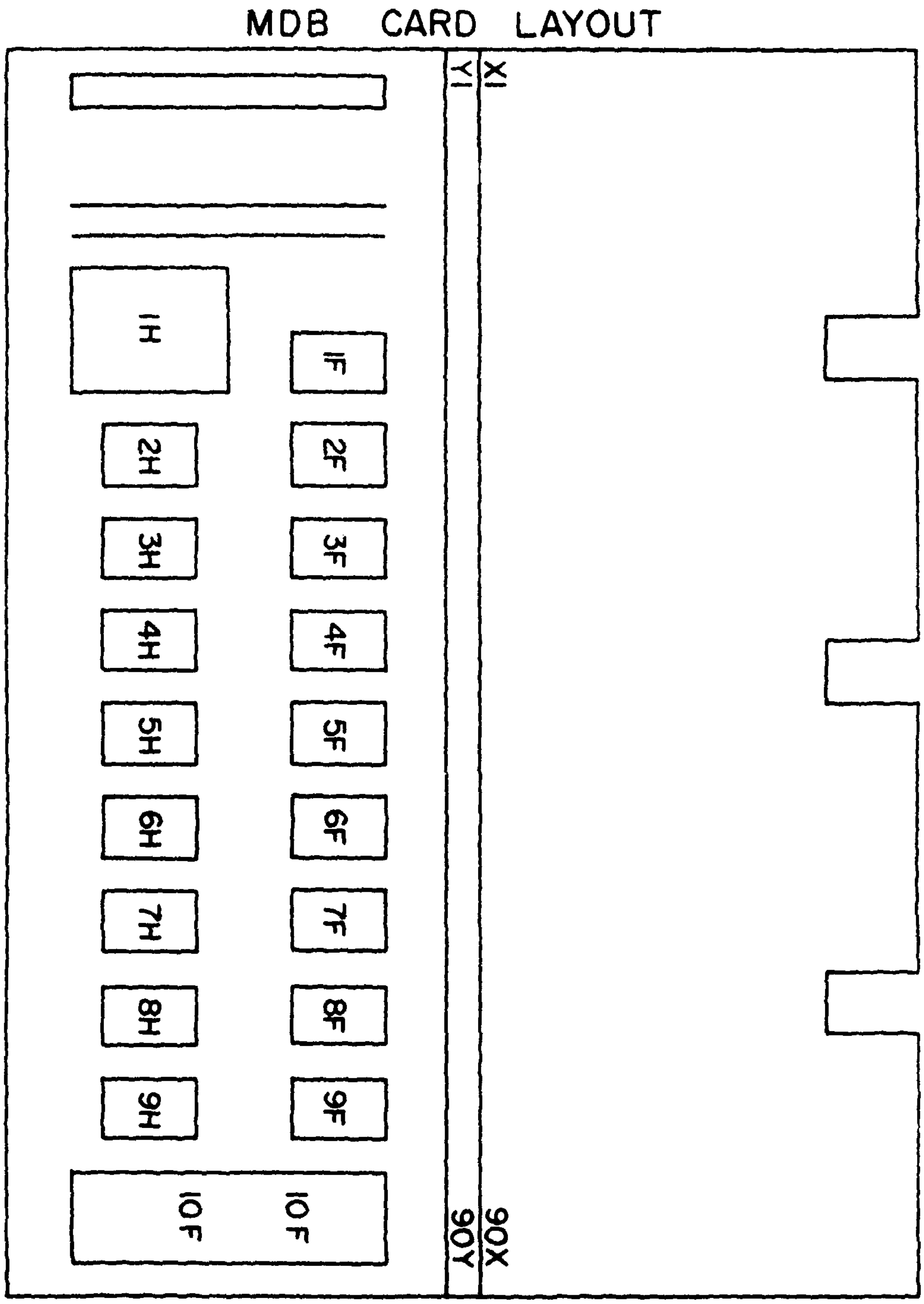


Fig. 7.6.
MDB-11C CARD LAYOUT

7.b.5.3. Jumpers List:

A list of the points that are jumpered for implementing the above mentioned combinations, for the address location as well as for the interrupt vector and the required priority level, is listed at the end of Appendix E.

8.0 SOFTWARE TIPS

8.1 General:

The structure of the interface cards (A/D Module and MDB general purpose interface card) leaves many options for use according to different softare orientations. For instance, the digitized information, processed by the A/D converter, can be routed into the processor in 2 ways: a) program controlled, i.e. the software has to check repeatedly the DONE bit in the status word, and after it is found to be 1, a data-read operation takes place. b) Interrupt controlled, i.e. the interface issues an interrupt call after a conversion process is completed. Another example is the mode of triggering the ADC. Since most of those hardware selections have software implications, software orientation should be indicated after the hardware selections have been made.

8.2 A/D Programming:

The software for the A/D readings is based on the following assumptions:

a) The DC channels should be read and their data updated once per a single video frame.

b) Updating should be completed as early as possible during the frame-scanning period.

c) The fastest way for processing all DC channels is to trigger a new conversion cycle as soon as the last data is read into the processor.

d) After all the channels are read into the processor, interrupt may be disabled until the next frame.

The flow chart in Figure 8.1 is based on the above-mentioned assumptions.

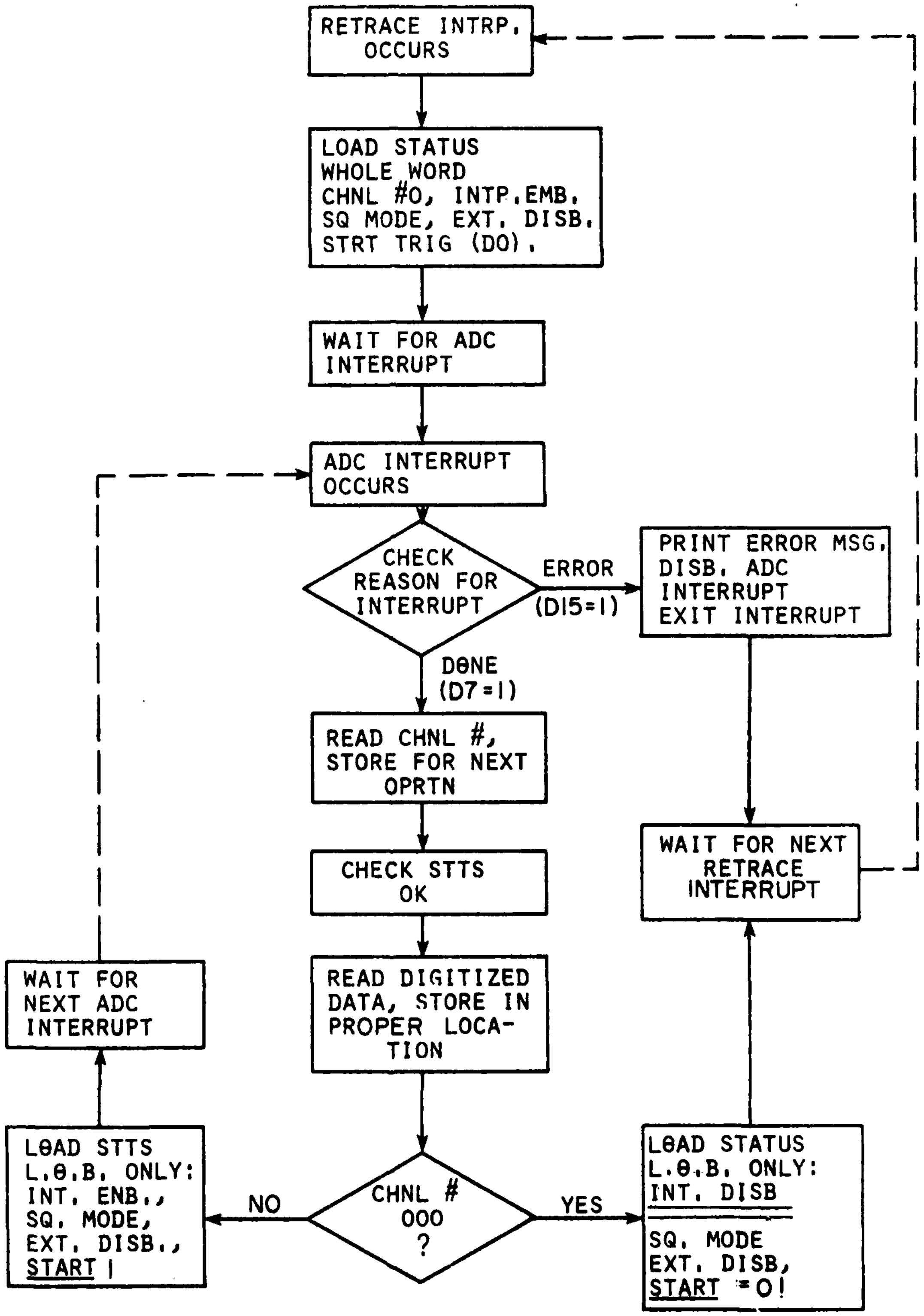

RETRACE INTRP, OCCURS
LOAD STATUS WHOLE WORD CHNL #0, INTP,EMB, SQ MODE, EXT, DISB, STRT TRIG (DO),
WAIT FOR ADC INTERRUPT
ADC INTERRUPT OCCURS
CHECK REASON FOR INTERRUPT
ERROR (D15=1)
PRINT ERROR MSG, DISB, ADC INTERRUPT EXIT INTERRUPT
DONE (D7=1)
READ CHNL #, STORE FOR NEXT OPRTN
CHECK STTS OK
READ DIGITIZED DATA, STORE IN PROPER LOCATION
WAIT FOR NEXT RETRACE INTERRUPT
WAIT FOR NEXT ADC INTERRUPT
CHNL # 000 ?
NO
YES
LOAD STTS L.O.B. ONLY: INT, ENB., SQ, MODE, EXT, DISB., START I
LOAD STATUS L.O.B, ONLY: INT, DISB
SQ, MODE EXT, DISB, START =O!

Fig. 8.1.
ADC SUBROUTINE

8.3 Keyboard Programming:

In the general purpose interface module there is no facility for enabling or disabling the interrupt. The interrupt logics is always enabled and there is no status register. Those facts make the programming a straight-forward task. The program has to wait for the interrupt call, and when it arrives, the only operation that has to take place is a read operation. The read operation should be proceeded by checking the status (High-order byte) for an error or malfunction, and if everything is OK, validate the read data. The writing operation, which means controlling the push-buttons illumination is independent of the interrupt and is entirely program-controlled.

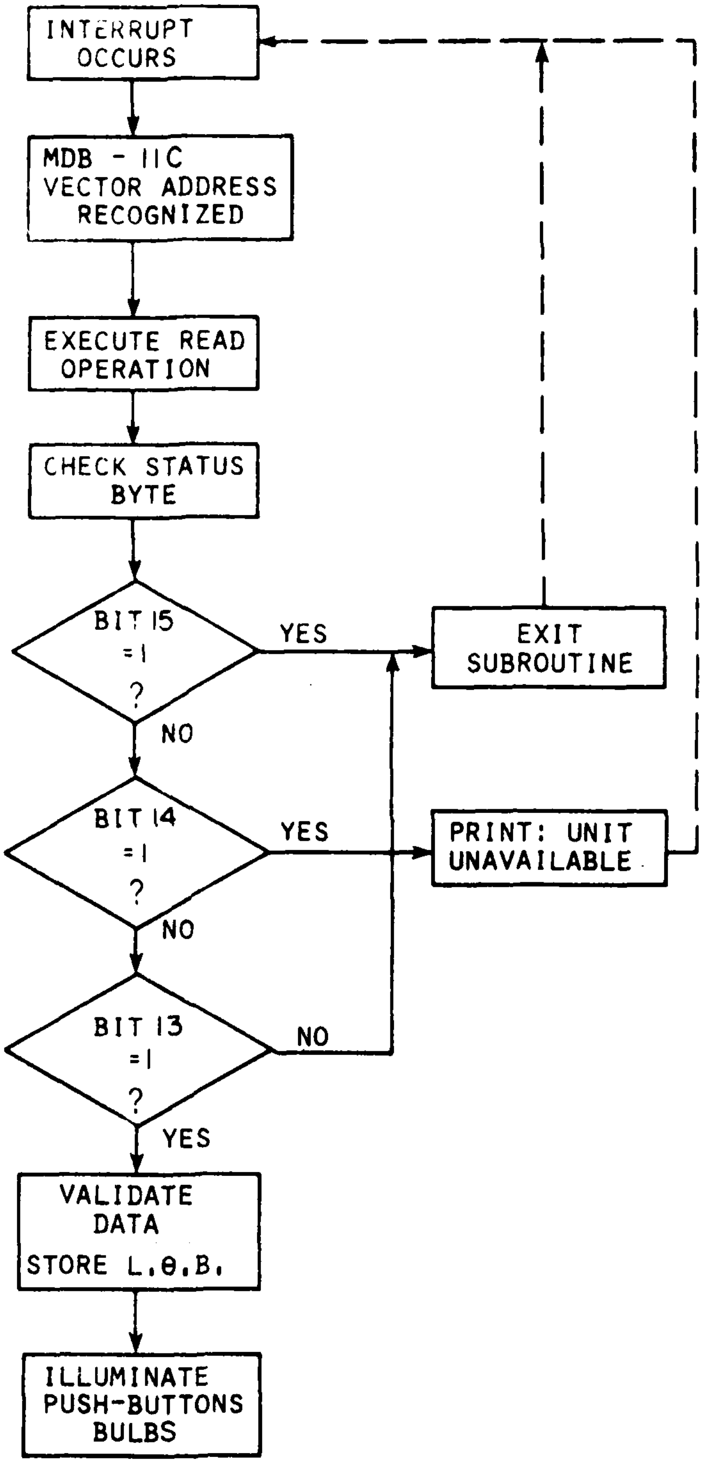
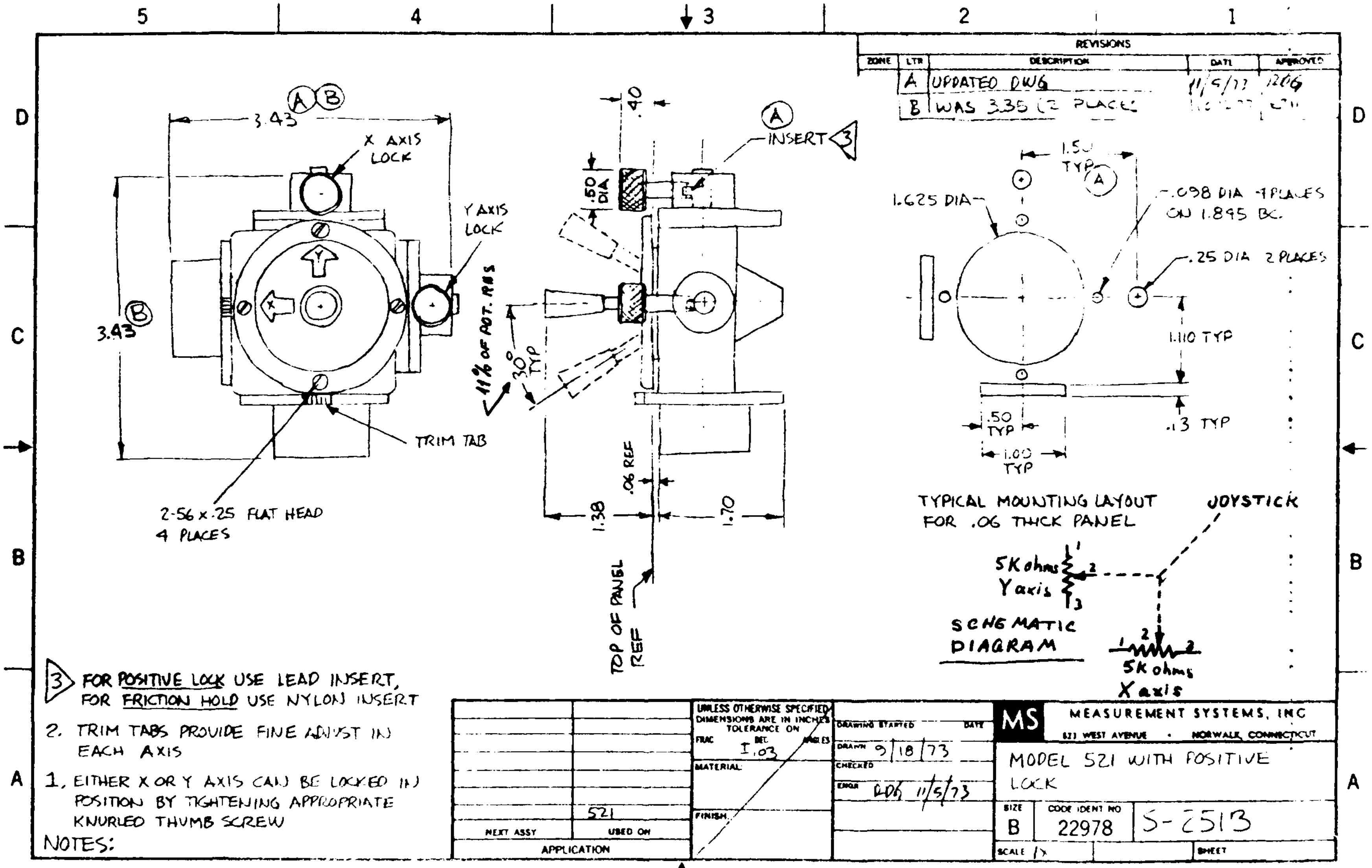


Fig. 8.2.
MDB-11C MODULE SUBROUTINE

APPENDIX A

Joystick Type 521-2513 Measurement Systems, Inc.

Data Sheet



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	A	UPDATED DWG	11/5/73	RDG
	B	WAS 3.35 (2 PLACES	[illegible]	[illegible]

3. FOR POSITIVE LOCK USE LEAD INSERT, FOR FRICTION HOLD USE NYLON INSERT

2. TRIM TABS PROVIDE FINE ADJUST IN EACH AXIS

1. EITHER X OR Y AXIS CAN BE LOCKED IN POSITION BY TIGHTENING APPROPRIATE KNURLED THUMB SCREW

NOTES:

	521
NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCE ON
FRAC | DEC ±.03 | ANGLES
MATERIAL:
FINISH:

DRAWING STARTED | DATE
DRAWN 9/18/73
CHECKED
ENGR RDG 11/5/73

MS MEASUREMENT SYSTEMS, INC
521 WEST AVENUE • NORWALK, CONNECTICUT

MODEL 521 WITH POSITIVE LOCK

SIZE B | CODE IDENT NO 22978 | S-2513
SCALE 1X | SHEET

APPENDIX B

Power Supply Data Sheets

28V - Powertec Type 22A-500

5V - Powertec Type 22B-100

POWERTEC

AN AIRTRONICS SUBSIDIARY

9138 DESOTO AVENUE
CHATSWORTH CALIFORNIA 91311
(213) 882-0004 TWX 910-494-2092

SUB-MODULAR D.C. POWER SUPPLY

MODULE 22A-500

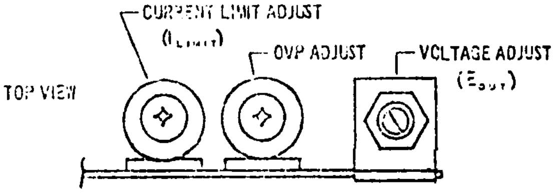
MODEL NUMBER—

ASSEMBLY NUMBER—

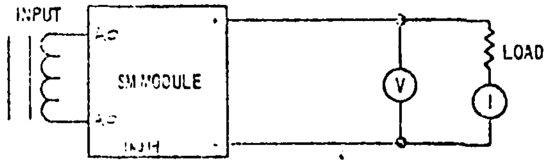
SPECIFICATIONS

WARNING: Requires power transformer to reduce line voltage to recommended input level.

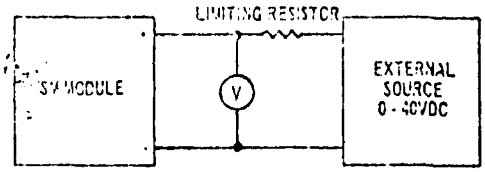
OUTPUT VOLTAGE	22.0 V to 30.0 V
OUTPUT CURRENT	.425 A @ 24 V - .400 A @ 28 V
LINE REGULATION	± .075 %
LOAD REGULATION	± .075%
RIPPLE PEAK-PEAK	<5MV
OVER CURRENT	50-130% of full rated load
OVER VOLTAGE	105-135% of ratings



POTIENTIOMETER LOCATION
FIGURE 1



CONNECTION DIAGRAM
FIGURE 2



OVP ADJUST SET-UP
FIGURE 3

ADJUSTMENT PROCEDURE

VOLTAGE ADJUST

Adjust output voltage to desired level at no load with unit connected as shown in figure 2. (Make sure OVP is in maximum clockwise position).

CURRENT LIMIT ADJUST

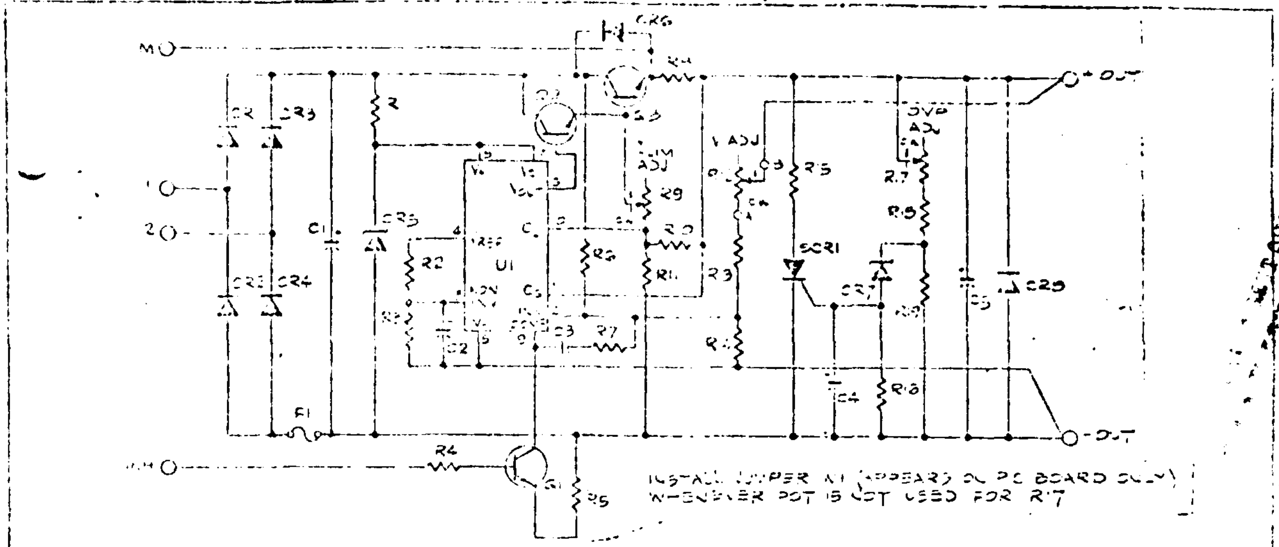
Adjust I_{LIMIT} to maximum clockwise position. Apply 125% of full load and adjust I_{LIMIT} until unit drops out of regulation 50 to 100mV.

CAUTION: DO NOT RUN UNITS OVER 5 MINUTES WITHOUT ADDITIONAL HEAT SINK

OVP ADJUST

Remove input power and load and apply an external voltage through a limiting resistor as shown in figure 3. Adjust OVP adjust until firing occurs at desired voltage as the external source is slowly increased. Select limiting resistor to limit current to 0.5ADC maximum after firing.

Refer to SM sales brochure PT3-9-72 for additional information and design parameters.



INSTALL JUMPER X1 (APPEARS ON PC BOARD ONLY) WHENEVER POT IS NOT USED FOR R17

REF DES	DESCRIPTION	VALUE	POWERTEC PART NO.
C1	CAP ELECT	[illegible]	32-13007-014
C2	CAP ELECT	1/50V	32-13005-001
C3	CAP FILM	.001, 100V	31-13005-0[illegible]
C4	CAP ELECT	1/50V	32-13005-001
C5	CAP ELECT	100, 50V	32-13005-007
CR1	DIODE	S1-2	11-13009-001
CR2	DIODE	S1-2	11-13009-001
CR3	DIODE	S1-2	11-13009-001
CR4	DIODE	S1-2	11-13009-001
CR5	DIODE ZENER	1N4738A	12-13015-026
CR6	DIODE	S1-2	11-13009-001
CR7	DIODE ZENER	1N[illegible]	12-13015-005
CR9	DIODE	S1-2	11-13009-001
R1	RES 1/2W 5%	750Ω	20-13004-070
R2	RES 1/2W 5%	2.2KΩ	20-13004-091
R3	RES	—	
R4	RES 1/2W 5%	47KΩ	20-13004-116
R5	RES 1/2W 5%	1KΩ	20-13004-073
R6	[illegible]	—	
R7	RES 1/2W 5%	3.3KΩ	20-13004-085
R8	RES 3W	1.2Ω	22-13014-014
R9	[illegible]	—	
R10	RES 1/2W 5%	3.9KΩ	20-13004-08[illegible]
R11	RES RN55C	1.62KΩ	21-13050-216

REF DES	DESCRIPTION	VALUE	POWERTEC PART NO.
R14	RES RN55C	357Ω	21-13050-[illegible]
R15	RES 5W	.47Ω	24-13014-00[illegible]
R16	RES 1/2W 5%	270Ω	20-13004-05[illegible]
R18	RES 1/2W 5%	1.3KΩ	20-13004-07[illegible]
R19	RES 1/2W 5%	680Ω	20-13004-0[illegible]
R9	POT	100Ω	23-13010-00[illegible]
R12	POT	1.5KΩ	23-13010-[illegible]
R17	POT	1.5KΩ	23-13010-[illegible]
SCR	SCR	10TF	13-13027-001
Q1	TSTR	[illegible]	[illegible]
Q2	TSTR	2N2102	[illegible]
Q3	TSTR	[illegible]	[illegible]
U1	I.C.	[illegible]	14-13042-001
F1	FUSE	1A	08-13012-[illegible]

Input fuse blows	(1) check fuse rating (2) possible overload (3) Ovp triggering with Q2, Q3, CR5 shorted (4) CR1, CR2, CR3, CR4, C1 shorted
Low output voltage, poor regulation, high ripple, loaded	(1) possible overload or current limit adj. R9 improperly adjusted (should be set for 120% of full load current prior to foldback) (2) possible Ovp triggering, check setting of R17 (3) U1 defective (4) CR5, R9, R3, C2, C3, R12, R13, C5, CR6 shorted (5) R1, Q2, Q3, R11, R2, R11 open
High output voltage, poor regulation, high ripple, loaded	(1) U1 defective (2) Q2, Q3, CR6, R2, R11 shorted (3) R12, R13, R3 open
High output voltage unloaded, OK loaded	(1) U1 defective (2) Q2, Q3 high leakage
Output noise	(1) U1 defective (2) C2, C5 open
Output oscillation	(1) U1 defective (2) C3, R7, C5 open
OVP triggers under normal operation	(1) check OVP setting (2) SCR1, CR7, R17, R18 shorted (3) C4, R19 open
OVP fails to trigger	(1) SCR1, CR7, R17, R18, R19 open (2) C4, R15 shorted
Inhibit does not function	(1) Q1, R4, R5 open
Excessive unit heating	(1) possible overload (2) inadequate heat sinking or heat sink bolted to uneven surface, no thermal compound used in heat sinking (3) input voltage too high

MODEL 22A-500

POWERTEC
AN AIRTRONICS SUBSIDIARY

9168 DESOTO AVENUE
CHATSWORTH CALIFORNIA 91311
(213) 882-0004 TWX 910-494-2092

SUB-MODULAR D.C. POWER SUPPLY

MODULE 22B-100

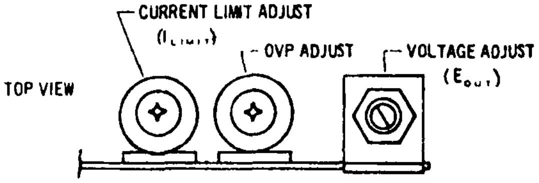
MODEL NUMBER—

ASSEMBLY NUMBER—

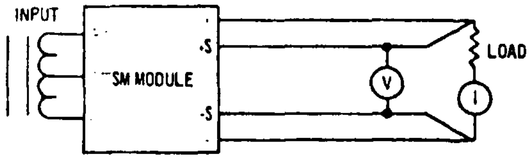
SPECIFICATIONS

WARNING: Requires power transformer to reduce line voltage to recommended input level.

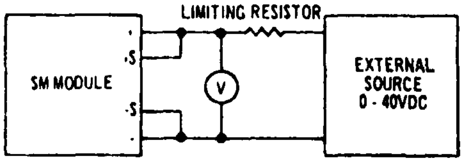
OUTPUT VOLTAGE	4.75 V to 7.0 V
OUTPUT CURRENT	3.0 A @ 5 V - 2.5 A @ 6 V
LINE REGULATION	± .075%
LOAD REGULATION	± .075%
RIPPLE PEAK-PEAK	<5MV
OVER CURRENT	50-130% of full rated load
OVER VOLTAGE	105-135% of ratings



POTIENTIOMETER LOCATION
FIGURE 1



CONNECTION DIAGRAM
FIGURE 2



OVP ADJUST SET-UP
FIGURE 3

ADJUSTMENT PROCEDURE

VOLTAGE ADJUST

Adjust output voltage to desired level at no load with unit connected as shown in figure 2. (Make sure OVP is in maximum clockwise position).

CURRENT LIMIT ADJUST

Adjust I_{LIMIT} to maximum clockwise position. Apply 125% of full load and adjust I_{LIMIT} until unit drops out of regulation 50 to 100mV.

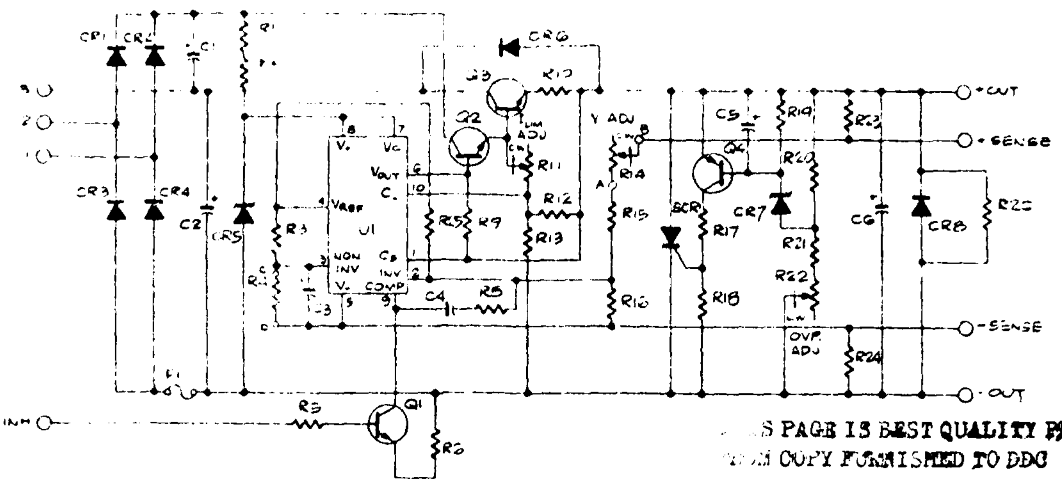
CAUTION: DO NOT RUN UNITS OVER 5 MINUTES WITHOUT ADDITIONAL HEAT SINK

OVP ADJUST

Remove input power and load and apply an external voltage through a limiting resistor as shown in figure 3. Adjust OVP adjust until firing occurs at desired voltage as the external source is slowly increased. Select limiting resistor to limit current to 0.5ADC maximum after firing.

Refer to SM sales brochure PT3-9-72 for additional information and design parameters.

17676-001



REF DES	DESCRIPTION	VALUE	POWERTEC PART NO.
C1	CAP ELECT	470/25V	32-13006-002
C2	CAP ELECT	9000/15V	32-13007-001
C3	CAP ELECT	1/50V	32-13006-001
C4	CAP FILM	.001/100V	31-13006-008
C5	CAP ELECT	1/50V	32-13006-001
C6	CAP ELECT	470/25V	32-13006-002
CR1	DIODE	S1-2	11-13009.001
CR2	DIODE	S1-2	11-13009.001
CR3	DIODE	3S1S	11-13010-001
CR4	DIODE	2S1S	11-13010-001
CR5	DIODE	–	
CR6	DIODE	S1-2	11-13009.001
CR7	DIODE, ZENER	IN751A	12-16774.006
CR8	DIODE	S1-2	11-13009-001
R1	RES, 1/2W, 5%	51Ω	20-13004-042
R2	RES, 1/2W, 5%	51Ω	20-13004-042
R3	RES, RN60C	3.01KΩ	21-13050-239
R4	RES, RN60C	4.02KΩ	21-13050 251
R5	RES, 1/2W, 5%	47KΩ	20-13004-113
R6	RES, 1/2W, 5%	1KΩ	20-13004-073
R25	RES	–	
R8	RES, 1/2W, 5%	3.3KΩ	20-13004-085
R9	RES, 1/2W, 5%	1KΩ	20-13004-073
R10	RES, BWH	.1Ω	24-13014-001

REF DES	DESCRIPTION	VALUE	POWERTEC PART NO
R12	RES	–	
R13	RES, 1/2W, 5%	510Ω	20-13004-066
R15	RES	JUMPER	
R16	RES, RN60C	1.5KΩ	21-13050-210
R17	RES, 1/2W, 5%	100Ω	20-13004-049
R18	RES, 1/2W, 5%	1KΩ	20-13004-073
R19	RES, 1/2W, 5%	1KΩ	20-13004-073
R20	RES, RN60C	1KΩ	21-13050-193
R21	RES	JUMPER	
R23	RES, 1/2W, 5%	10Ω	20-13004-025
R24	RES, 1/2W, 5%	10Ω	20-13004-025
R26	RES	–	
R11	POT	100Ω	23-13016-004
R14	POT	1.5KΩ	23-13032-006
R22	POT	500Ω	23-13016-007
SCR1	SCR	2N4441	13-13015-001
Q1	TSTR	2N2222A	10-13019.002
Q2	TSTR	13159-2	10-13159-002
Q3	TSTR	19655-1	10-19655-001
Q4	TSTR	2N2907A	10-13020-002
U1	I.C.	723CE	14-13034-001
F1	FUSE	8A	03-16045-011

Input fuse blows	(1) check fuse rating (2) possible overload (3) Ovp triggering with Q2, Q3, CR6 shorted (4) CR1, CR2, CR3, CR4, C1, C2 shorted
Low output voltage, poor regulation, high ripple, loaded	(1) possible overload or current limit adj. R11 improperly adjusted (should be set for 120% of full load current prior to foldback) (2) possible Ovp triggering, check setting of R22 (3) U1 defective (4) C1, CR5, R4, C3, Q1, C4, R11, R14, R15, CR8, C6 shorted (5) R1, R2, R3, R13, R16 open
High output voltage, poor regulation, high ripple, loaded	(1) U1 defective (2) Q2, Q3, CR6, R3, R16 shorted (3) R4, R14, R15 open
High output voltage unloaded, OK loaded	(1) U1 defective (2) Q2, Q3 high leakage
Output noise	(1) U1 defective (2) C3, C6 open
Output oscillation	(1) U1 defective (2) C4, R8, C6 open
OVP triggers under normal operation	(1) check Ovp setting (2) SCR1, Q4, CR7, R21, R22 shorted (3) C5, R20 open
OVP fails to trigger	(1) SCR1, R17, Q4, CR7, R21, R22 open (2) R18, R19, C5 shorted
Inhibit does not function	(1) Q1, R5, R6 open
Excessive unit heating	(1) possible overload (2) inadequate heat sinking or heat sink bolted to uneven surface, no thermal compound used in heat sinking (3) input voltage too high

APPENDIX C

Test-Box

Description and Schematics

The test-box described in this appendix is designed to be connected to the unit instead of the interface cards through two cable mounted connectors: CC1 connected to J1 (DC connector) and CC2 connected to J2 (Logic connector). The test-box has no built in power source, and has to be furnished by the DC supply from the unit under test. For this purpose, and for the supply voltage measurement purpose, all the supply voltages, +28V, -28V, 28VGND, +5V and 5VGND appear in the rear panel connectors J1 and J2, although they are not routed further onto the interface during current regular operation except of the 2 grounds, that have to be routed to the interface for common grounding purposes. The DC section contains the selector switch and the two binding-posts for connecting a voltmeter or a DVM. The selector can select each of the 3 supply voltages and each of the 4 pots and the 2 joysticks output (each axis of each joystick), total number of 11 positions.

The logic section test circuits contain display indicators for the encoded bits and a flip-flop to record the Device-Call pulse, which is too narrow (100 μSec) to observe directly. The SHIFT signal is used for resetting this flip-flop. The recorded DVC signal is also displayed on an LED indicator, as well as the EOM and the power status signal.

Besides, there is a separate push-button to test the illumination bulbs in the 16 keys that operate by externally received signals.

Since the joystick unit has a built-in power supply, it is recommended to turn its power off before connecting or disconnecting the unit to or from the test box.

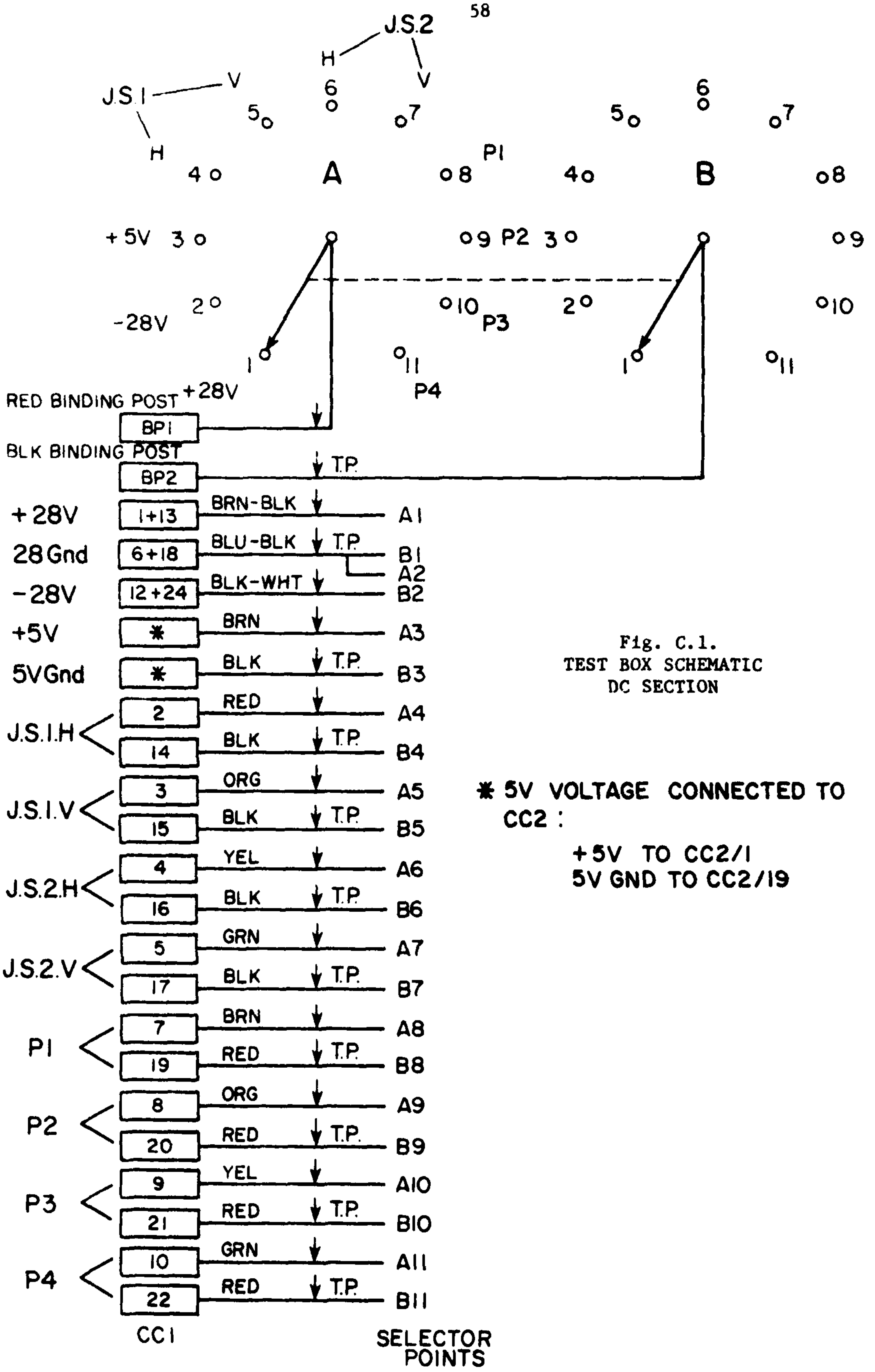


Fig. C.1.
TEST BOX SCHEMATIC
DC SECTION

✱ 5V VOLTAGE CONNECTED TO CC2 :

+5V TO CC2/1
5V GND TO CC2/19

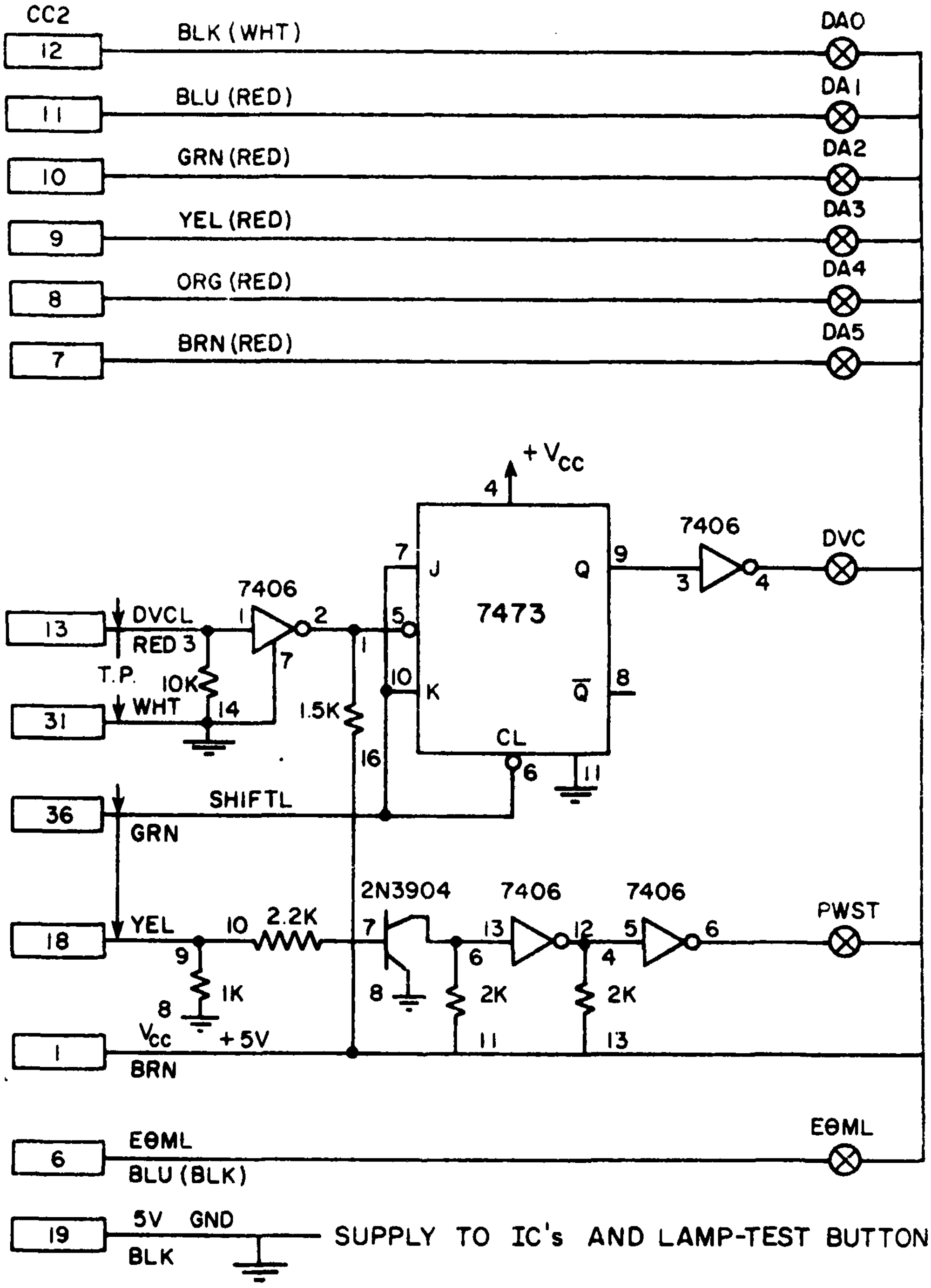


Fig. C.2.
TEST BOX SCHEMATIC
DIGITAL SECTION

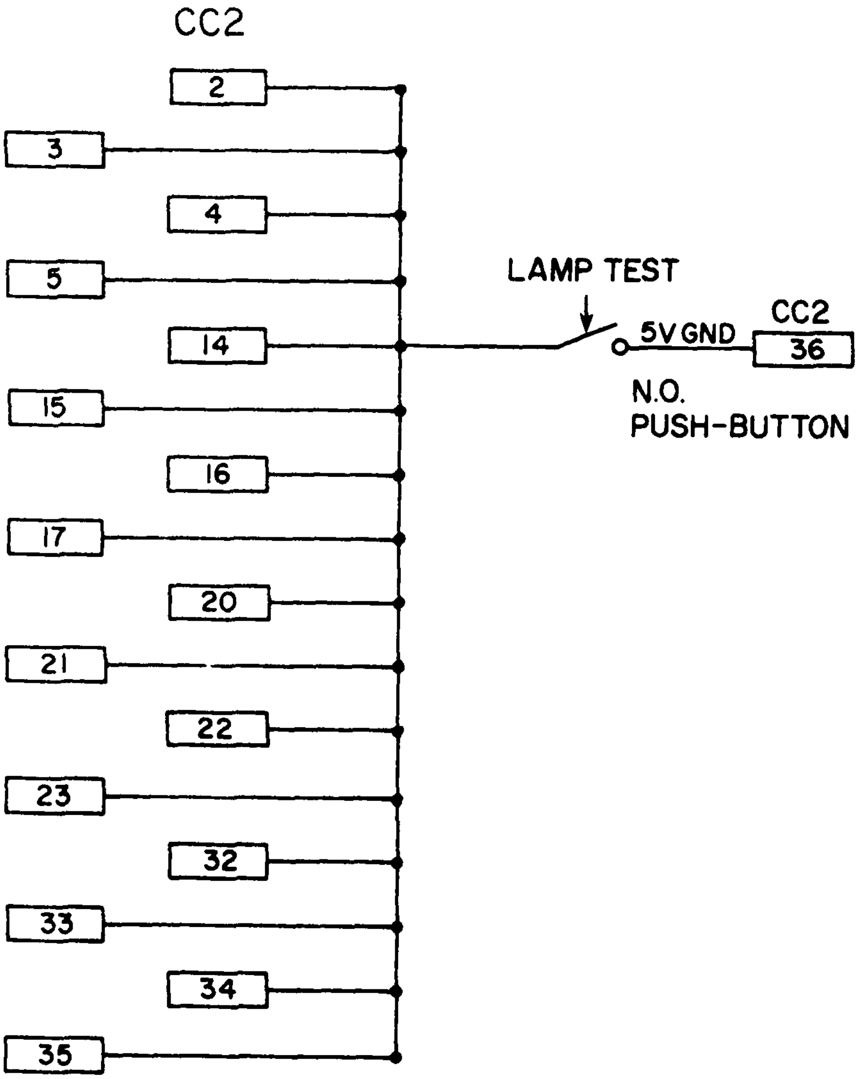


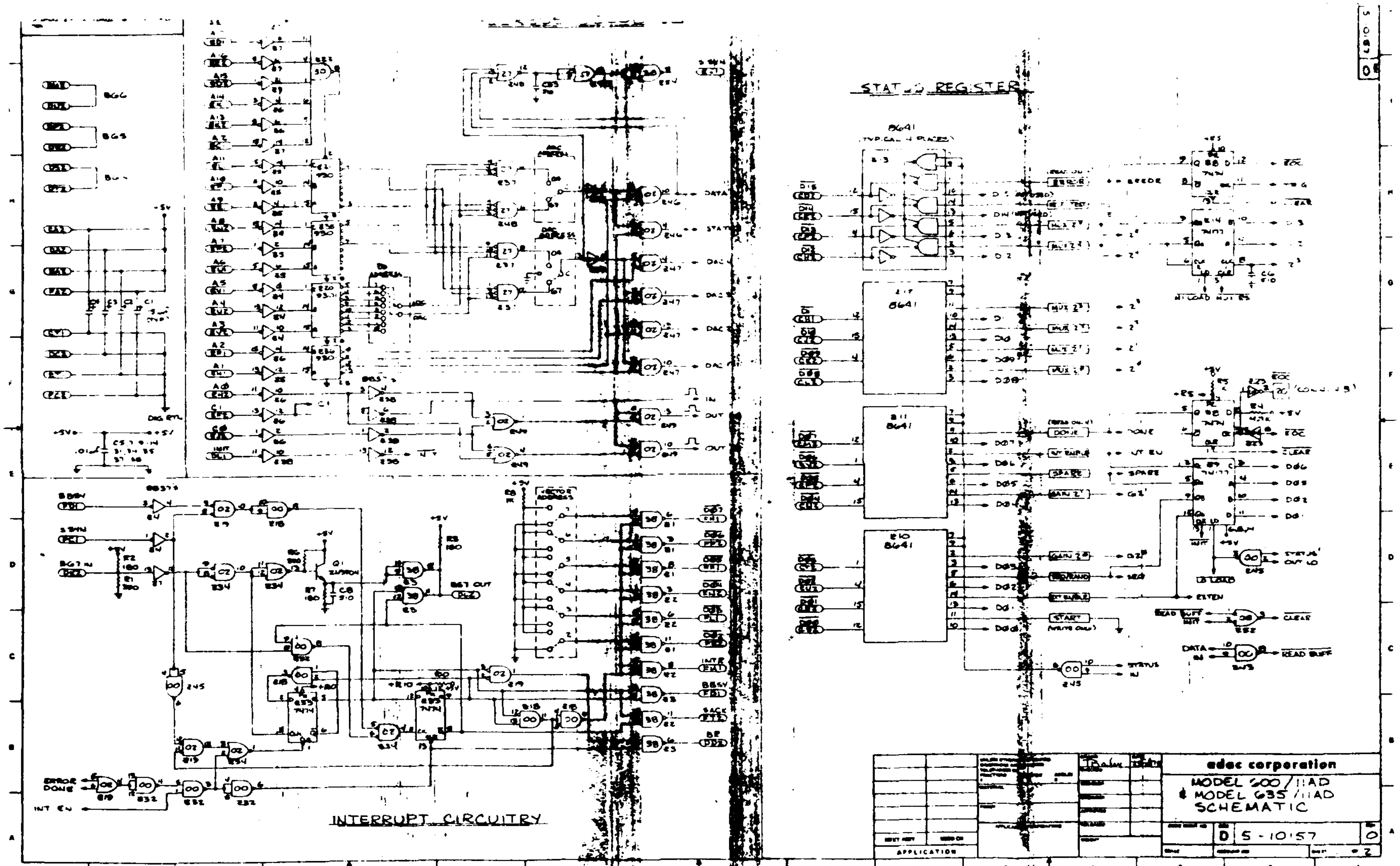
Fig. C.3.
TEST-BOX SCHEMATIC
LAMP-TEST CIRCUIT

APPENDIX D

Analog to Digital Converter

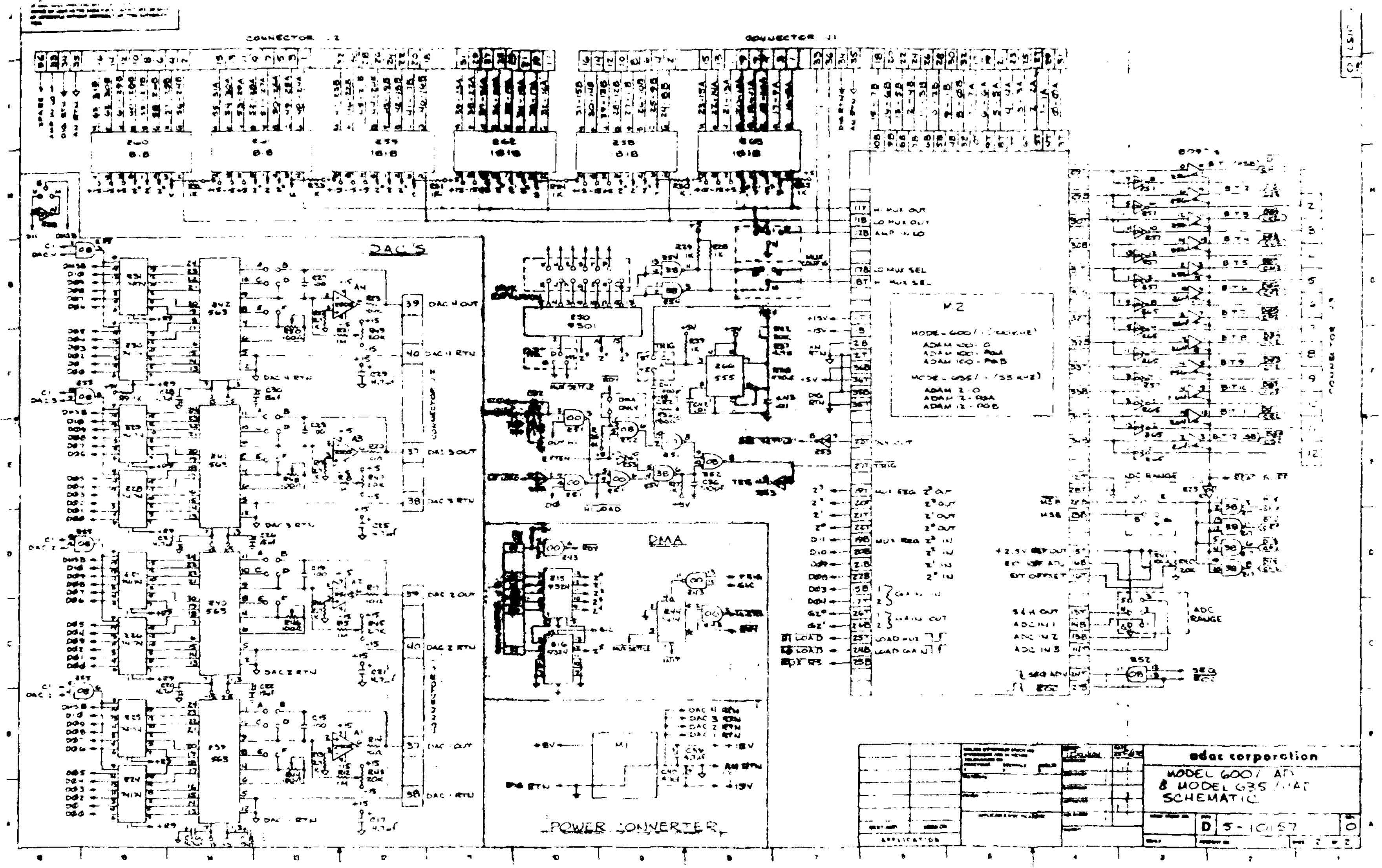
Model 635/11AD

Manufactured by ADAC Corporation



62

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


CONNECTOR J2
CONNECTOR J1
DAC'S
DMA
POWER CONNERTER
K2
MODEL 600/ (50 KHZ)
ADAM 100: 0
ADAM 100: P6A
ADAM 100: P6B
MODEL 635/ (55 KHZ)
ADAM 2: 0
ADAM 2: P6A
ADAM 2: P6B
HI MUX OUT
LO MUX OUT
AMP IN LO
LO MUX SEL
HI MUX SEL
TRIG
ADC RANGE
ADC RANGE
S&H OUT
ADC IN 1
ADC IN 2
ADC IN 3
SEG ADV
DAC 4 OUT
DAC 4 RTN
DAC 3 OUT
DAC 3 RTN
DAC 2 OUT
DAC 2 RTN
DAC 1 OUT
DAC 1 RTN
555
9301
565
+5V
+15V
-15V
APPLICATION
adac corporation
MODEL 600/ AD
& MODEL 635 /AT
SCHEMATIC
D
5-10157
0
2 OF 2

APPENDIX E

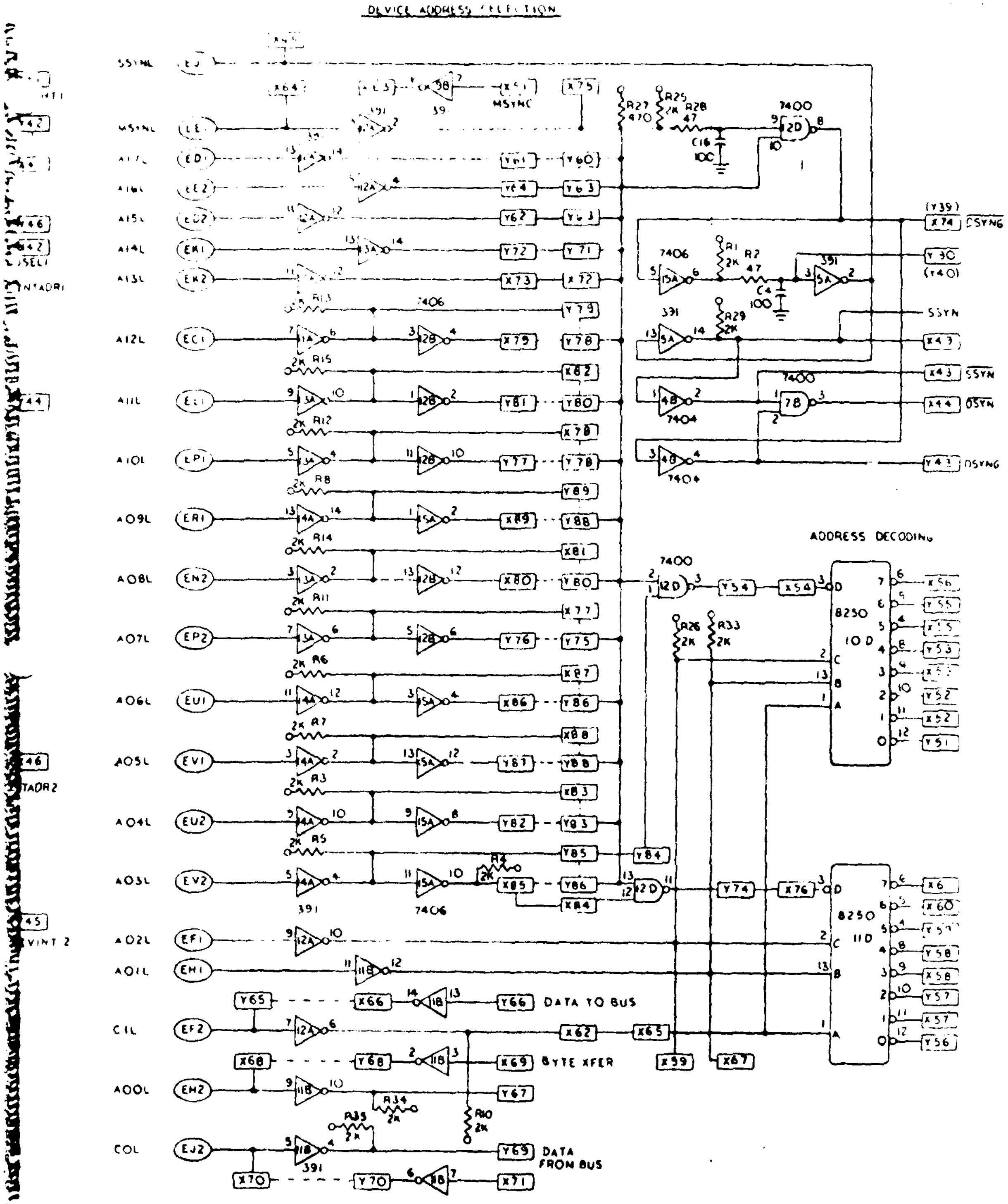
MDB-11C

General Purpose Interface Module

The module described in this appendix is built on a single card and serves as a general purpose interface. In this application it is used for transfer of keyboard information and controlling keyboard illumination. The logic on this card has inlet points, to which the logic described in chapter 7.b is connected. Connections can be identified by the numbered points.

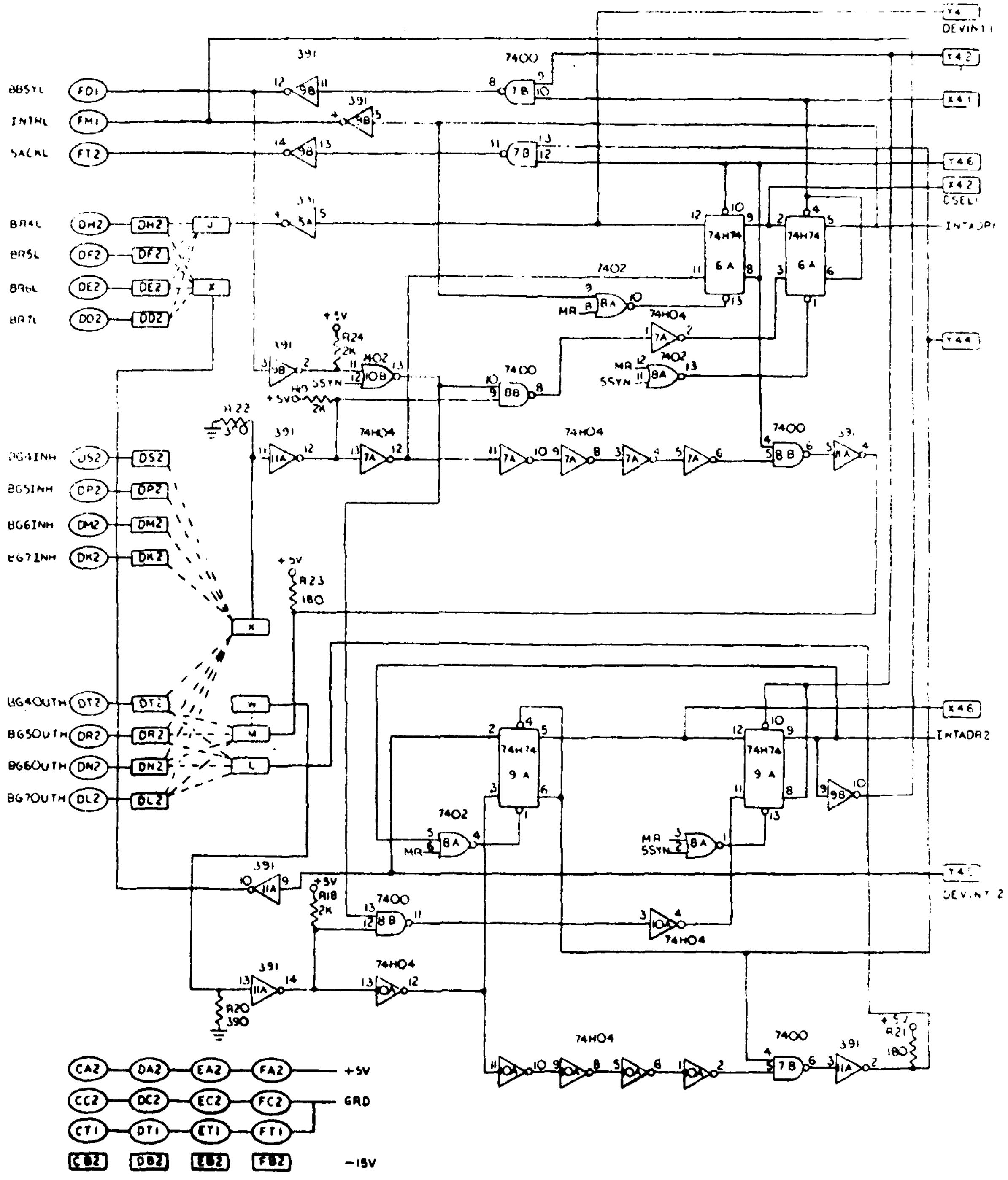
Apart from the users interface, some internal connections jumpers had to be striped, according to software considerations. The card has no DMA capability, and requests bus-control for issuing interrupt calls only. It has the capability of issuing two separate interrupt calls, assigning a separate vector address for each one and a capability of requesting bus control on a different or equal priority levels. In this case only one interrupt level is used (DEVINT1), and the bus-request and bus grant take place on level #4 (BR4-BG4). Jumpers are striped accordingly.

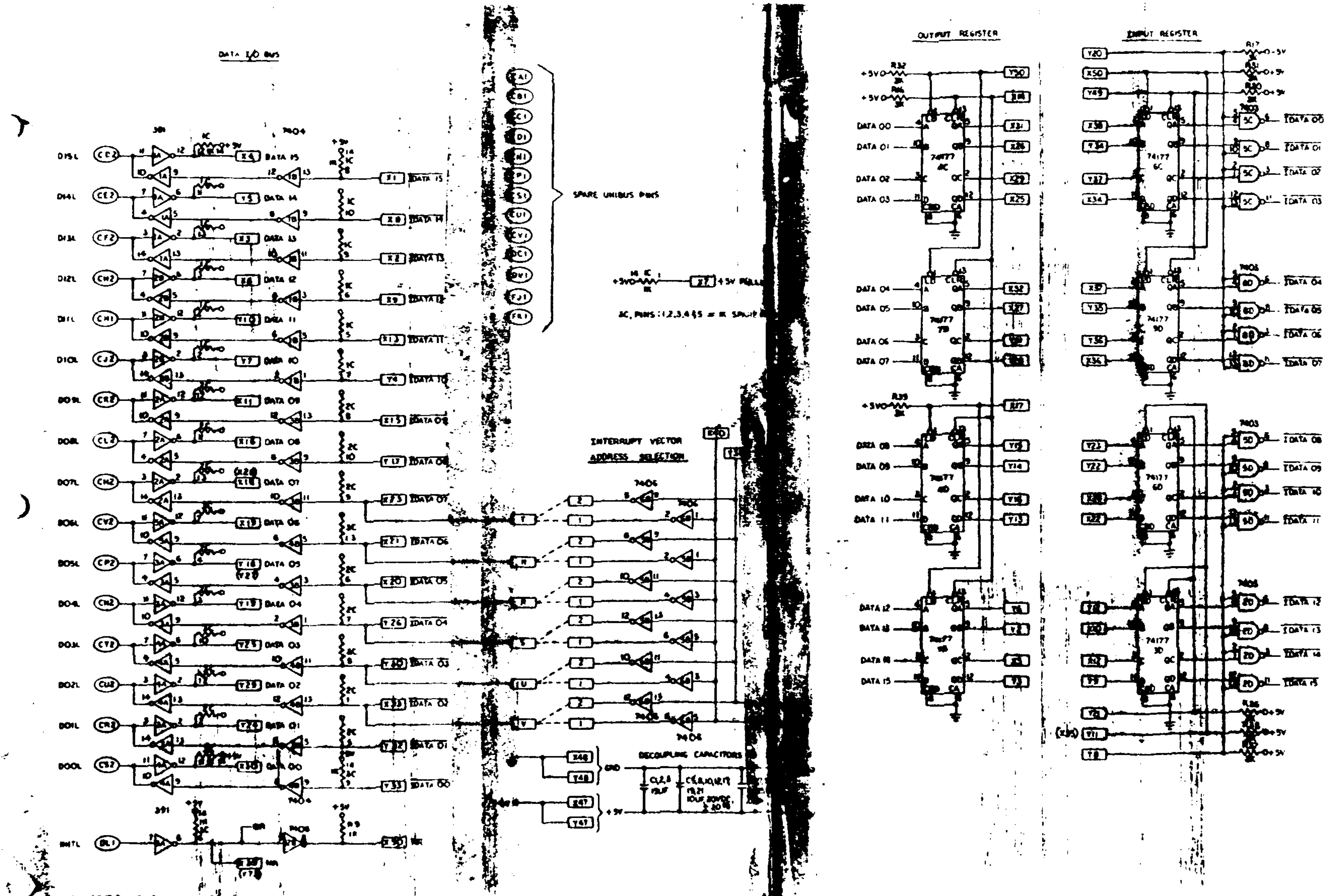
On the following page jumpers connection lists are specified. More details about the module can be found in the MDB-11C module Instruction manual.



MDB-11C Logic Diagram, Sheet 1

INTERRUPT CONTROL





MDB-11C Logic Diagram, Sheet 2

MDB-11C JUMPERS LIST

SIGNAL	POINT	TO POINT
A03	x85	y86
A04	x83	y83
A05	y87	y88
A06	x86	y86
A07	y75	y76
A08	y80	x80
A09	y88	x89
A10	y78	y77
A11	y80	x82
A12	y78	x79
$\overline{\text{IDATA02}}$	V	1
$\overline{\text{IDATA04}}$	S	1
$\overline{\text{IDATA06}}$	N	1
$\overline{\text{IDATA07}}$	T	1
BR4L	DH2	J
BG4INH	DS2	K
BG4OUTH	DT2	M
BG5INH-BG5OUTH	DP2	DR2
BG6INH-BG6OUTH	DM2	DN2
BG7INH-BG7OUTH	DK2	DL2

APPENDIX F

Electronic Board
Wiring List
and
Layout Diagram

IC PIN #

	SIGNAL MNEMONIC				LOGIC SECTION									
	ROD00L	PC 36	U9 15	U4 4										
	ROD01L	PC 35	U9 14	U4 1										
	ROD02L	PC 88	U9 13	U4 13										
	ROD03L	PC 89	U9 12	U4 10										
	ROD04L	PC 38	U9 11	U3 4										
	ROD05L	PC 39	U9 10	U3 1										
	ROD06L	PC 40	U9 9	U3 13										
	ROD07L	PC 90	U9 8A	U3 10										
	ROD08L	PC 82	U9 8	U2 13										
	ROD09L	PC 81	U9 7	U2 10										
	ROD10L	PC 83	U9 6	U2 1										
	ROD11L	PC 29	U9 5	U2 4										
	ROD12L	PC 21	U9 4	U1 13										
	ROD13L	PC 30	U9 3	U1 10										
	ROD14L	PC 19	U9 2	U1 1										
	ROD15L	PC 18	U9 4	U1 1										
	DAT000	U4 6	D0 0	U10 1										
	DAT001	U4 3	D0 1	U10 3										
	DAT002	U4 11	D0 2	U10 5										
	DAT003	U4 8	D0 3	U10 13										
	DAT004	U3 6	D0 4	U10 11										
	DAT005	U3 3	D0 5	U10 9										
	DAT006	U3 11	D0 6	U10 1										
	DAT007	U3 8	D0 7	U11 3										
	DAT008	U2 11	D0 8	U11 5										
	DAT009	U2 8	D0 9	U11 13										
	DAT010	U2 3	D0 10	U11 11										

I.C.
PIN#

	SIGNAL MNEMONIC													
	DAT011	U2 6	DO 11	U11 9										
	DAT012	U1 11	DO 12	U12 1										
	DAT013	U1 8	DO 13	U12 3										
	DAT014	U1 3	DO 14	U12 5										
	DAT015	U1 6	DO 15	U12 13										
	A	U10 2	PC 17											
	B	U10 4	PC 16											
	C	U10 6	PC 15											
	D	U10 12	PC 14											
	E	U10 10	PC 13											
	F	U10 8	PC 12											
	G	U11 2	PC 11											
	H	U11 4	PC 10											
	J	U11 6	PC 9											
	K	U11 12	PC 8											
	L	U11 10	PC 7											
	M	U11 8	PC 6											
	N	U12 2	PC 5											
	P	U12 4	PC 4											
	R	U12 6	PC 3											
	S	U12 12	PC 2											
	DA03L	U6 3	PC 42											
	DA02L	U6 13	PC 41											
	GND	U6 4	PC 47											
	DA23G	U6 4	U6 8											
	DA01G	U6 8	PC 46											
	DA01L	U6 7	PC 94											

I C
PIN#

	SIGNAL MNE MONIC													
	DA00L	U6 9	PC 95											
	PWL	U15 6	PC 34											
	DVCL	U5 3	PC 43											
	DA05L	U5 13	PC 92											
	DA04L	U5 7	PC 91											
	EOML	U5 9	PC 93											
	GND	U5 4	U5 8											
	DA5DVCG	U5 4	PC 45											
	DA4EOMQ	U5 8	PC 44											
	(pull-up P.U. resistor)	U15 7	U6 15	U5 1	U5 15	U5 11								
	$\overline{\text{DA03}}$	U6 2	DI 3	U17 9										
	$\overline{\text{DA02}}$	U6 14	DI 2	U16 6										
	$\overline{\text{DA01}}$	U6 6	DI 1	U16 7										
	$\overline{\text{DA00}}$	U6 10	DI 0	U16 9										
	$\overline{\text{DVC}}$	U5 2	DI 7	U7 12										
	$\overline{\text{DA05}}$	U5 14	DI 5	U17 6										
	$\overline{\text{DA04}}$	U5 6	DI 4	U17 7										
	$\overline{\text{EOM}}$	U5 10	DI 6	U21 8										
	SHIFTL	U15 5	PC 37											
		U7 7	U8 3	U8 4										
		U7 6	U8 13											
		U8 1	U8 2	U7 15										
		U8 15	U7 14											
	DLVC	U7 9	U7 13	U21 1										
	$\overline{\text{DLVC}}$	U7 4	U7 10											
	PWRST	U7 3	U7 11	C1 +	U15 4	U23 5								
	$\overline{\text{CINH}}$	U21 13	U7 2											

IC / PIN#

	SIGNAL MNEMONIC													
		U23 8	U7 1											
	EGS	U23 10	U23 11											
	EOM	U21 9	U23 9											
	GS1	U16 14	U23 13											
	GS2	U17 14	U23 12											
	Ei1	U20 6	U16 5	U23 1										
	Ei2	U20 8	U17 5	U23 2										
	EiT	U23 3	U22 13	U21 7	U23 4									
	$\overline{\mathrm{EiT}}$	U22 12	U21 14	U21 2										
	EOMRST	U23 6	U21 6											
	GND	U21 11	U21 10											
	EOMS	U21 5	U15 3	PC 45	U8 9									
	Vcc	U8 16	U8 14											
	AR	PC 52	U13 1	U18 2										
	AP	PC 53	U13 2	U16 10	U5 5	U6 5								
	BR	PC 54	U13 3	U18 3										
	BP	PC 55	U13 4	U16 11										
	CR	PC 56	U13 5	U18 5										
	CP	PC 57	U13 6	U16 12										
	DR	PC 58	U13 7	U18 6										
	DP	PC 59	U13 8	U16 13										
	ER	PC 60	U13 9	U18 8										
	EP	PC 61	U13 10	U18 1										
	FR	PC 62	U13 11	U18 9										
	FP	PC 63	U13 12	U16 2										
	GR	PC 64	U13 13	U18 11										
	GP	PC 65	U13 14	U16 3										

	SIGNAL MNEMONIC													
	HR	PC 66	U13 18	U18 12										
	HP	PC 67	U13 1	U16 4										
		U18 1	U20 1											
		U18 4	U20 2											
		U18 10	U20 4											
		U18 13	U20 5											
	JR	PC 68	U14 1	U19 2										
	JP	PC 69	U14 2	U17 10	U6 1	U6 11								
	KR	PC 70	U14 3	U19 3										
	KP	PC 71	U14 4	U17 11										
	LR	PC 72	U14 5	U19 5										
	LP	PC 73	U14 6	U17 12										
	MR	PC 74	U14 7	U19 6										
	MP	PC 75	U14 8	U17 13										
	NR	PC 76	U14 9	U19 8										
	NP	PC 77	U14 10	U17 1										
	PR	PC 78	U14 11	U19 9										
	PP	PC 79	U14 12	U17 2										
	RR	PC 80	U14 13	U19 11										
	RP	PC 31	U14 14	U17 3										
	SR	PC 32	U14 15	U19 12										
	SP	PC 33	U15 2	U17 4										
		U19 1	U20 13											
		U19 4	U20 12											
		U19 10	U20 10											
		U19 13	U20 9											

	SIGNAL MNEMONIC													
	+28V	PC 48	U32 11	U32 12	R2V 3	R2H 3	R1H 3	R4V 3	U31 11	U31 12				
	-28V	PC 49	U32 1	U32 2	U32 14	U32 10	U31 14	U31 10	U31 2	U31 1				
		U31 6	RV1U 3											
		U31 5	RV2U 3											
		U31 16	RV1L 3											
		U31 15	RV2L 3											
		U32 5	RV3L 3											
		U32 6	RV4L 3											
		U32 16	RV4L 3											
		U32 15	RV3L 3											
	RV1U	PC 27	RV1U 2											
	RV1L	PC 28	RV1L 2											
	RV2U	PC 25	RV2U 2											
	RV2L	PC 26	RV2L 2											
	R1V	PC 84	R1V 2											
	R1H	PC 86	R1H 2											
	R5	PC 85	U31 7											
	R6	PC 87	U31 3											
	R2H	PC 98	R2H 2											
	R2V	PC 96	R2V 2											
	R7	PC 99	U32 7											
	R8	PC 97	U32 3											
	RV3U	PC 23	RV3U 2											
	RV3L	PC 24	RV3L 2											
	RV4U	PC 20	RV4U 2											
	RV4L	PC 22	RV4L 2											

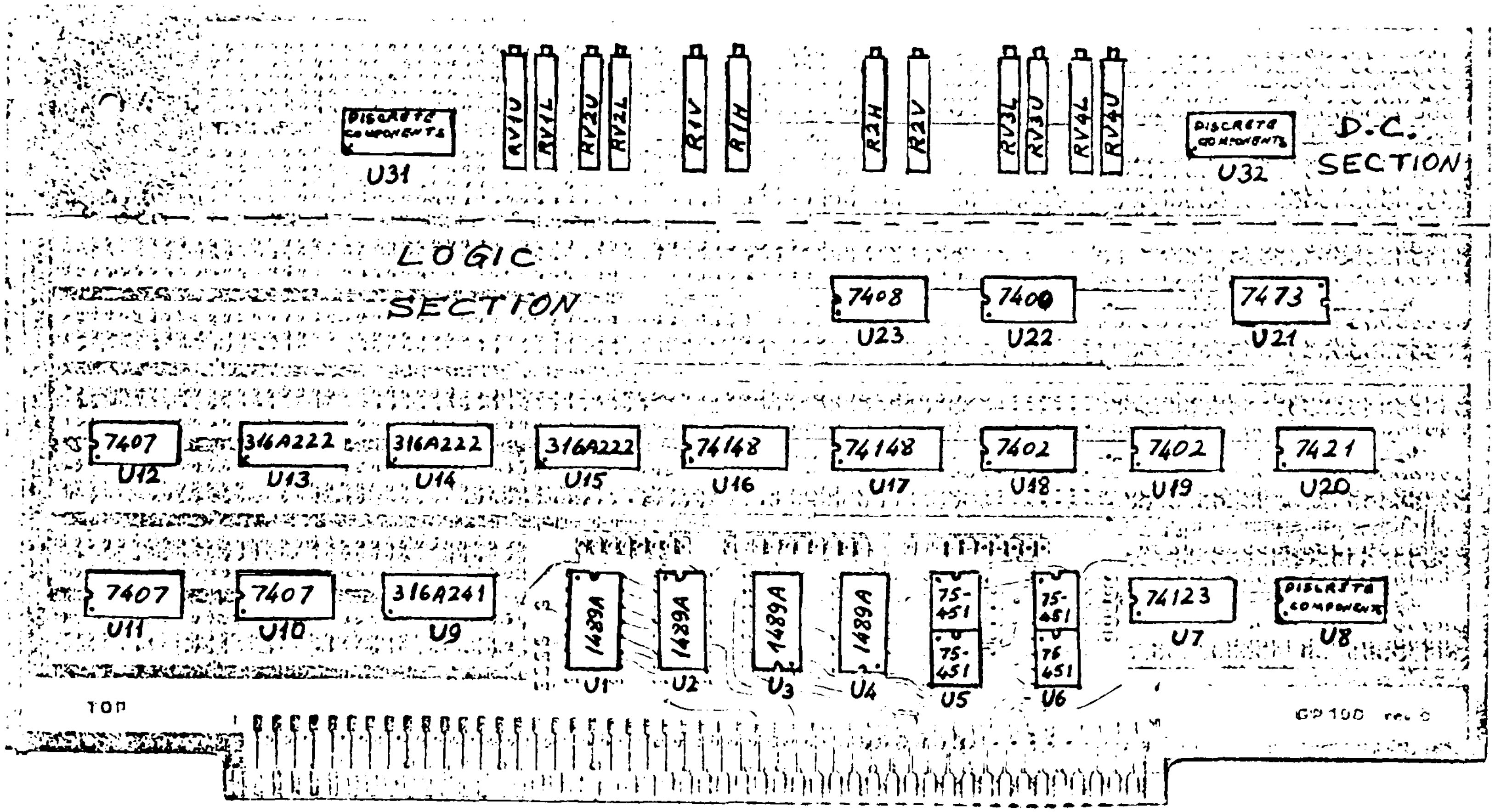


Fig. E.1 Electronic Board Layout

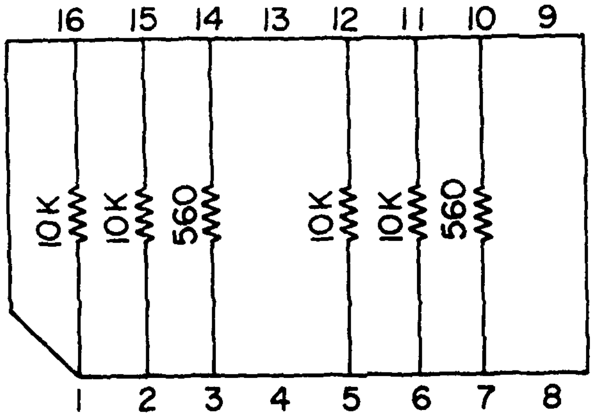


Fig. F.2. DISCRETE COMPONENTS ADAPTER FOR DC SECTION

AD-A090 234 COLORADO STATE UNIV FORT COLLINS DEPT OF ATMOSPHERI--ETC F/G 4/2
DIGITAL IMAGING RESEARCH FOR METEOROLOGICAL APPLICATIONS.(U)
AUG 80 T A BRUBAKER, T H VONDER HAAR DAAG29-76-G-0324

UNCLASSIFIED ARO-13971.7-GS NL

2 OF 2
AD
A090234

END
DATE
FILMED
11-80
DTIC

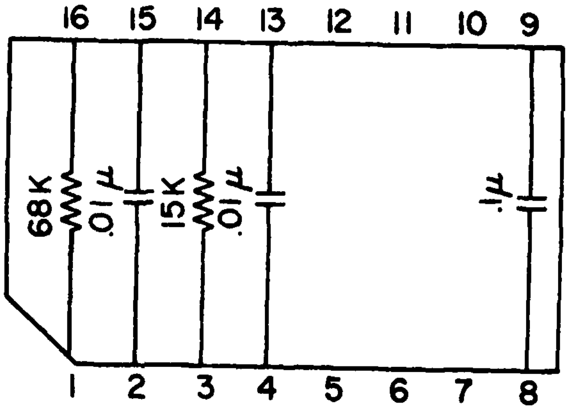


Fig. F.3.
DISCRETE COMPONENT ADAPTER FOR
LOCATION U8

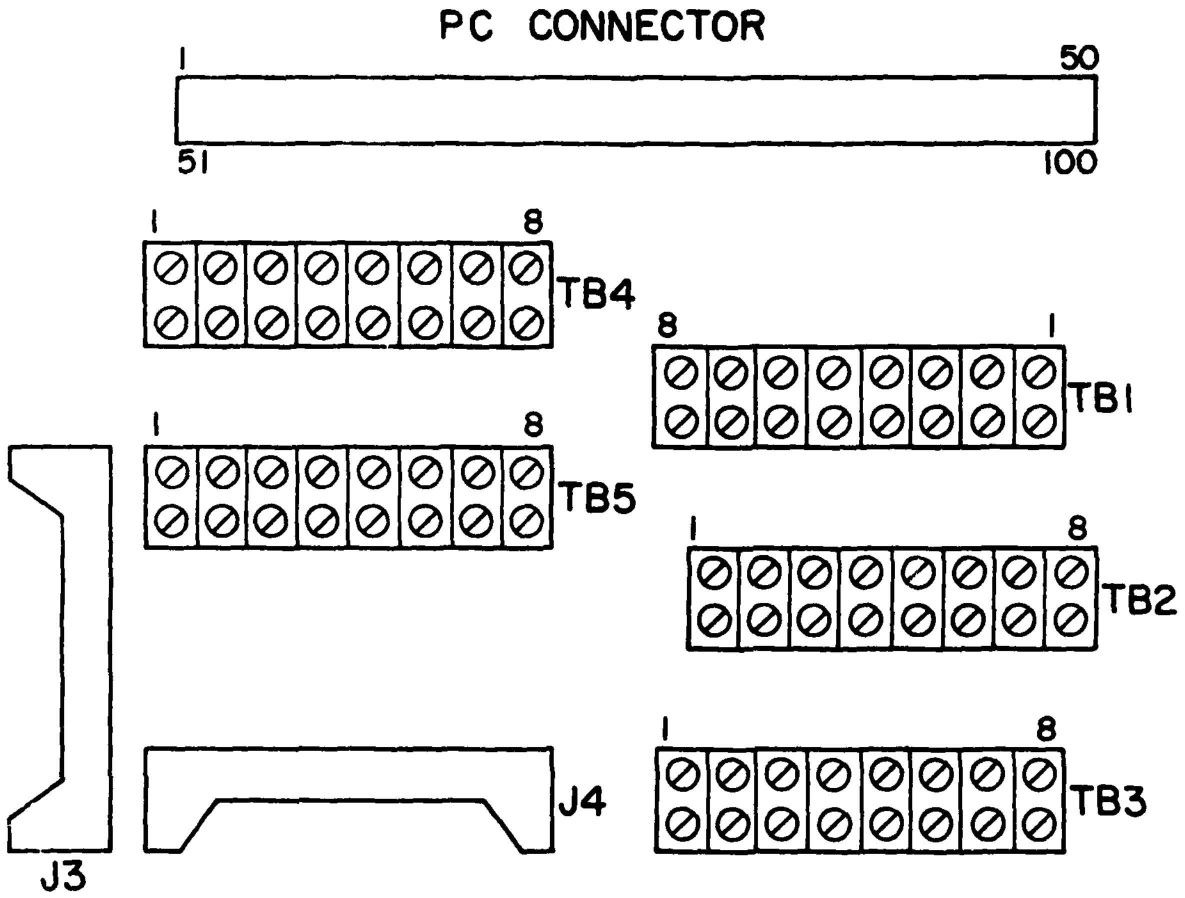


Fig. F.4.
BOTTOM JUNCTION PANEL LAYOUT
BOTTOM VIEW

BOARD PINS ASSIGNMENT

Pin #	Signal	Pin #	Signal	Pin #	Signal
1	+5V	37	SHIFTL	73	LP
2	S	38	ROD04L	74	MR
3	R	39	ROD05L	75	MP
4	P	40	ROD06L	76	NR
5	N	41	DA02L	77	NP
6	M	42	DA03L	78	PR
7	L	43	DVCL	79	PP
8	K	44	U5G	80	RR
9	J	45	EOMS	81	ROD09L
10	H	46	U6G	82	ROD08L
11	G	47		83	ROD10L
12	F	48	+28V	84	R1V
13	E	49	-28V	85	R5
14	D	50	GND	86	R1H
15	C	51	+5V	87	R6
16	B	52	AR	88	ROD02L
17	A	53	AP	89	ROD03L
18	ROD15L	54	BR	90	ROD07L
19	ROD14L	55	BP	91	DA0rL
20	RV4U	56	CR	92	DA05L
21	ROD12L	57	CP	93	EOML
22	RV4L	58	DR	94	DA01L
23	RV3U	59	DP	95	DA00L
24	RV3L	60	ER	96	R2V
25	RV2U	61	EP	97	R8
26	RV2L	62	FR	98	R2H
27	RV1U	63	FP	99	R7
28	RV1L	64	GR	100	GND
29	ROD11L	65	GP		
30	ROD13L	66	HR		
31	RP	67	HP		
32	SR	68	JR		
33	SP	69	JP		
34	PWL	70	KR		
35	ROD01L	71	KP		
36	ROD00L	72	LR		

APPENDIX D
SOFTWARE DOCUMENTATION

PROGRAM C

Purpose: To print the current (X,Y) cursor size.

Usage under OS.
   Type: C or @C

Author: Marvin Brown

System-Language: ADVISAR - FTN4

Routines Used: RMPAR, FRCUR, LURQ, CLTTY, EXIT

Example:

```
>C
>CURSOR SIZE (X,Y) = 103, 27
>
```

PROGRAM CLMEM

Purpose:
To set a frame to a given value.

Usage:

```
   under RTE II or FRGR:  RU, CLMEM,,f,u

   under OS:  @CLMEM or *RU,CLMEM,,f,n
     where - f = the frame to be cleared
             u = two times the value that is to be in the frame.

     note - when using the @ under OS, - f is the frame being displayed
                                         on the RED signal system
                                       - n is the X cursor position.
```

Author:
Marvin Brown

System-Language:
ADVISAR-FTN4

Routines Used:
RMPAR, IAND, MEMIO, EXIT

Example:

PROGRAM COPY

Purpose:
To copy data between mag tapes.

Usage:
Under RTE II or FMGR

```
RU, COPY, mode, fr, lu1,lu2,u

Where:  mode = B for binary or C for coded
          fr = F for files or R for records
         lu1 = where the data is coming from
         lu2 = where the data is going to
           u = the number of transfers to be performed
```

Author:
Marvin Brown

System-Language:
RTE II - FRN4

Routines Used:
RMPAR, EXEC, ISUNT, REIO, EXIT

Example:

```
RU, COPY, B, R, 13, 8, 5
```

Copies five binary records from 9-track 1600 BPI to the HP 7-track drive.

PROGRAM D

Purpose:

To output the data value from the current frame at the current cursor location.

Usage:

Under RTE II or FMGR

```
RU, D,,f,x,y
```

Under OS

```
D or @D or *RU,D,,f,x,y
```

Where: f = the frame to interrogate
x = the x position
y = the y position

Note: under OS using D or @D, the current frame is the frame on the RED signal system and x,y is the current cursor position.

Author:

Marvin Brown

System-Language:

ADVISAR-FTN4

Routines Used:

RMPAR, FRCUR, MEMIO, LURQ, CLTTY EXIT

Example:

```
>D
>DATA VALUE = 127
>
```

PROGRAM DC

Purpose:
This program prompts for cursor parameters. Cursor position, size, type and color can be set.

Usage:
Under RTE II or FMGR
RU, DC
Under OS
@DC

Note: - null parameter entered do not change that parameter of the cursor
- cursor position and size flags are turned off.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
FMPAR LUCON, RSCOM, LURQ, CLTTY, WSCOM, EXIT

Example:

```
>@DC
>ENTER x, y POSITION (0-511, 0-511) - 100, 110
>ENTER x, y SIZE (0-511, 0-511) - 21, 54
>ENTER type (0-8), BW (0-4), RGB (0-8) - 1, 3, 4
>
```

PROGRAM DEMO

Purpose:
This program transfers frame file(s) from 9-track 1600 BPI tape to memory frame(s).

Usage:

```
under RTE-II or FMGR
  RU, DEMO,,f,n

under OS
  @DEMO or *RU,DEMO,,f,n

where:  f = the first frame to be transferred to
        n = the number of frames to transfer

note:  - in OS, if @DEMO is typed, the current frame which is being
         displayed on the RED signal system is used for f.  Also, n
         is one.
       - default values for f and n are 0 and 8, respectively if RU,
         DEMO is used.
```

Author:
Marvin Brown

System-Language:
ADVISAR-FTN4

Routines Used:
RMPAR, WSCOM, BFINP, ISUNT, MEMIO, EXIT

Example:

PROGRAM DF

Purpose:
   To prompt for any display frame changes.

Usage:
   Under OS:
     @DF

   Note: - null parameters entered do not change the frame displayed.
          - current display frames are used and output with the prompt.
          - see also DFO for overlay frames and turning on or off a signal system.

Author:
   Marvin Brown

System-Language:
   ADVISAR - FTN4

Routines Used:
   RMPAR, FRCUR, LURQ, CLTTY, RSCOM, WSCOM, EXIT

Example:

```
>@DF
>ENTER DISPLAY FRAMES - (0,1,2,2,3) - ,,1,1
>@DF
>ENTER DISPLAY FRAMES - (0,1,1,1,3) - 1,1,1,1,1
>
```

PROGRAM DFO

Purpose:
To prompt for any frame or overlay changes.

Usage:
Under RTE II or FMGR:
RU,DFO

Under OS:
@DFO

Note:
- the signal systems to be changed are prompted for.
- all signal systems are changed in the same way.
- the current setup of the first signal system entered is output in the prompt.

Author:
Marvin Brown

System Language:
ADVISAR - FTN4

Routines Used:
RMPAR, LUCON, LURZ, CLTTY, MOD, RSCOM, IAND, ISHFT, IOR, WSCOM, EXIT

Example:

PROGRAM DIG

Purpose: To digitize a picture and transfer it into the Intel memory.

Usage:

```
under RTE-II or FMGR
  RU, DIG,,f
under OS
  @DIG or *RU, DIG,,f
```

where: f = the frame to be transferred to

note: in OS, if @DIG is typed, the current frame which is being displayed on the RED signal system is used for f.

Author: Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:

RMPAR, IAND, LODXY, EXEC, MEMIO, EXIT

PROGRAM DLP

Purpose:
To prompt for loop parameters. The parameters include loop key, number of frames in the loop, loop rate, dwell and the loop sequence.

Usage:
Under RTE II or FMGR:
RU, DLP

Under OS:
@DLP

Note: null parameter entered do not change that parameter.

Author:
Marvin Brown

System-Language:
ADVISAR-FTN4

Routines Used:
RMPAR, LUCON, LURQ, CLTTY, MOD, RSCOM, EXIT

Example:

```
>@DLP
>ENTER SIGNAL SYSTEM(S) - (0-4) - 0,3
>KEY COUNT RATE DWELL 1 2 3 4 5 6 7 8
> 0    8    10    5   0 1 2 3 4 5 6 7
>-,,5,2
>
```

PROGRAM DSCR

Purpose: To interrogate and prompt for the current image screen position.

Usage:

```
under RTE-II or FMGR -
  RU, DSCR
under OS
  @DSCR
```

Note: screen coordinates are bounded by 0 and 511.

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, LUCON, RSCRN, LURQ, CLTTY, BOUND, WSCRN, EXIT

```
 EXAMPLE:
>@DSCR
>ENTER START X, END X, START Y, END Y
           0     511          0     511 - 100,200,100,200
>@DSCR
>ENTER START X, END X, START Y, END Y
         100     200        100     200 - 400,600
>@DSCR
>ENTER START X, END X, START Y, END Y
         400     511        100     200 - ,,0,511
>@DSCR
>ENTER START X, END X, START Y, END Y
         400     511          0     511 - 0
>@DSCR
>ENTER START X, END X, START Y, END Y
           0     511          0     511 -
>
>
>
>
>
>
>
>
```

PROGRAM EQ

Purpose: To prompt for parameters to define a signal system enhancement.

Usage:

```
Under RTE-II or FMGR -
     RU, EQ
Under OS -
     @EQ
```

Note:
- parameters are prompted for until an illegal signal system is entered
- parameters are described later.

Author:

Marvin Brown

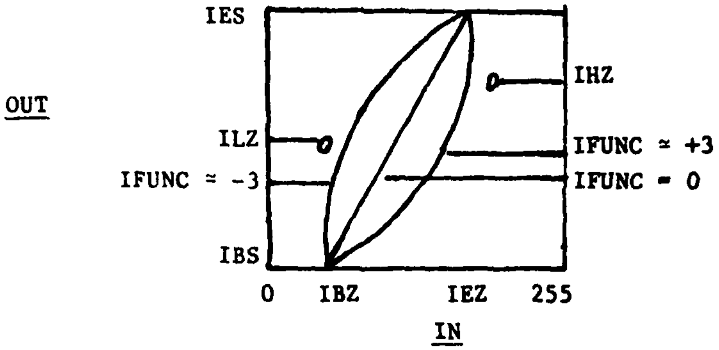
System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, LUCON, LURQ, CLTTY, EXEC, BOUND, FLOAT, ALOG, EXP, ENHEO, IFIX

EQ Parameter description
(IEN, IBZ, IEZ, IFUNC, ILZ, IHZ, NSP, IBS, IES)



IEN = The signal system to define.
IBZ = The starting value to be ehanhced
IEZ = The ending value to be enhanced
IFUNC = The enhancement function
= 0 a linear function, i.e. IFUNC (IBZ) = IBS linear to IFUNC (IEZ) = IES
> 0 then exponential between IBZ and IEZ. This highlights numbers close to IEZ (bright areas)
< 0 then a logarithmic function is used on values between IBZ and IEZ. This highlights numbers close to IBZ (dark areas)
ILZ = The value to which the input values less than IBZ are set.
IHZ = The value to which the input values greater than IEZ are set.
NSP = The number of steps output that are in the enhancement between IBZ and IEZ.
IBS = The value output when a lookup is done on IBZ.
IES = The value output when a lookup is done on IEZ.

PROGRAM EV

Purpose:

To define enhancement segments

Usage:

Under RTE-II or FMGR -
RU, EV
Under OS,
@EV

Note: The parameters are prompted for until an illegal signal system is entered.

Author:

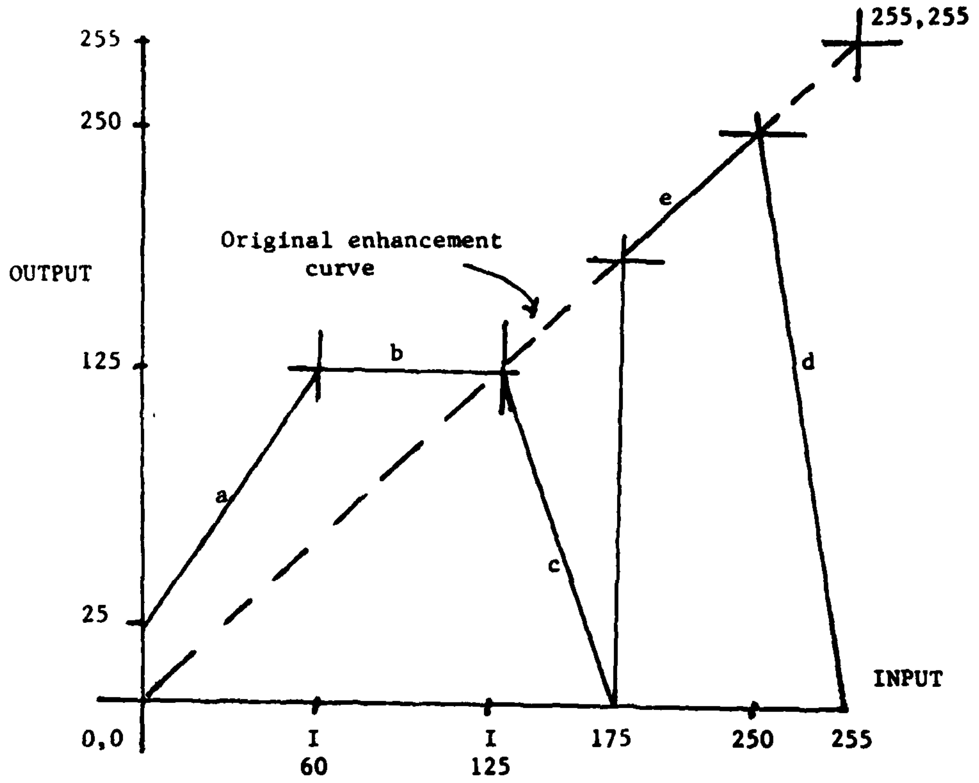
Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, LUCON, NHVCV, EXEC, LURQ, CLTTY, BOUND, IFIX, FLOAT, INHIO



INPUT TO TTY	COMMENTS
0, 0 60, 25, 125	SEGMENT a
0, 61, 125, 125, 125	SEGMENT b
0, 126, 175, 125, 0	SEGMENT c
0, 250, 255, 250, 0	SEGMENT d
9	TERMINATE SEQUENCE

NOTE: SEGMENT e COMES FROM ORIGINAL ENHANCEMENT CURVE

PROGRAM JOYZM

Purpose:
To enlarge the date contained in the dursor to the full size of another frame.

Usage:
Under OS:

```
@JOYZM
```

Note: - JOYZM prompts for a frame to write into.
- the current frame being displayed on the RED signal system is enlarged
- the current position and size of the cursor is used for the zoom.

Author:
Marvin Brown

System Language:
ADVISAR - FTN4

Routines Used:
RMPAR, LUCON, IAND, IABS, FLOAT, INT, LURQ, CLTTY, EXIT, MEMIO

Example:

```
>@JOYZM
>ZOOM FRAME 0 TO -7
```

PROGRAM MOVE

Purpose: To transfer a frame to another frame in the Intel memory.

Usage:

```
Under RTE or FMGR
     RU, MOVE,,rf
Under OS
     @MOVE or *RU, MOVE,,f
```

Where: f = the frame to transfer from.

Note:
- MOVE prompts for the destination frame
- in OS, if @MOVE is typed, the current frame which is being displayed on the RED signal system is used for f.
- if an illegal frame is entered or *BR,MOVE is typed, MOVE is aborted.

Author: Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, LUCON, IAND, LURQ, CLTTY, MEMIO, IFBRU, EXIT

PROGRAM N

Purpose:
To initialize the ADVISAR system. The frame looping is reset, the cursor is reset, and the enhancement tables are set to their locations (i.e. a linear enhancement).

Usage:
Under RTE II or FMGR:
RU, N

Under OS
N

Note:
Cursor size and position keys are turned off.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
WSCOM, ENHIO, MEMIO, EXIT

Example:

PROGRAM NRMLZ

Purpose: To normalize the digitizing system.

Usage: Under RTE-II or FMGR

```
RU, NRMLZ
```

Under OS

```
@NRMLZ
```

Note: The user is prompted to perform tasks (see example)

Author: Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RMPAR, LURQ, CLTTY, STTUS, KEYIN, AJSET, LDZRO, LDRNG, CRSXY, CLOSS, AJZRO, LODXY, SAMPL, QCALD, EXIT, EXEC, LDSEF, DIGUL, IUNPK, IABS, FLOAT

PROGRAM ØS

Purpose: This program provides a convenient interface between the user and various image manipulation routines, navigation programs and the RTEII operating system. Three basic command types are available for the user.

One letter keystrokes activate immediate image routines such as setting the looping flag, controlling the cursor, and printing out single data values. ØS can also initiate more complex programs if the program name is prefixed by an @ key. Examples are five manipulation, navigation and graphic output. If an * is pressed, then any RTEII command can be entered. Any program that is running can be stopped using this command by typing * BR, program name or *OF, program name.

To run under TREII or FMGR, Type: RU,ØS

ONE LETTER KEYINS

LETTER	TYPE	FUNCTION
A	Int	Advances frame position
B	Int	Backs up frame position
C	Ext	Print cursor size
D	Ext	Print data value
E		
F		
G	Int	Select cursor color
H		
I		
J		
K	Ext	Print loop parameters
L	Int	Start and stop loop
M		
N	Ext	Initialize display
O	Int	Select cursor type
P	Int	Start and stop position joystick
Q	Int	Return to RTE
R	Int	Reset to beginning of loop
S	Int	Start and stop size joystick
T	Ext	Print TV coordinate
U		
V		
W		
X		
Y		
Z		

@ Programs

Name	Description
CLMEM	Clear the displayed memory.
DC	Prompt for cursor parameters.
DEM	Transfer a frame file to a frame.
DF	Prompt for display frame parameters.
DFO	Prompt for frame overlay parameters.
DLP	Prompt for loop parameters.
JOYZM	Enlarge the data in the cursor.
SCOM	Read and write ADVISAR system common.
SDEMO	Save a frame file on mag tape.
VIDIO	Change the video output format.
ENHCE	Do joystick enhancement.

Errors:

1) if a one letter key in is not defined, it is either ignored and another prompt is output (i.e. 7 # 7 where # was the undefined character.) or "NO SUCH PROG" is printed.

2) if an @ program has been called and the program is not under the RTE II system, "NO SUCH PROG" is displayed.

3) if an RTE II command is entered with an * , then RTEII errors are output.

Author:
   Marvin Brown

System-Language:
   ADVISAR FTN4

Routines Used:
   RMPAR, LUCON, VURQ, EXEC, REIO, DEUBU, ISHFT, RSCOM, IOR, IAND, MESSS, WSCOM.

Internal Routines:
   INITO, KINTR, CRPAR, RUNPR, FRCON, ROTCO, LOOPC, ROTTP, JOYST, QUIT, JOYS, AT, COLON, STAR.

Core-Timing:

Examples:

PROGRAM BFUNC

Purpose: This program does define finary frame operations on two frames and writes the result in a third frame. The presently defined operations are: AV - the point average.
DF - the point difference.
MN - the point minimum.
MX - the point maximum.

```
To run under RTE II or RMGR, type:  RU, BFUNC,,fn,f1,f2,fw

    or under OS, type:  *RU, FBUNC,,fu,f1,f2,fw

                        where - fn = the two letter
                                f1 = the first frame
                                f2 = the second frame
                                     (in DF, fw=f1-f2)
                                fw = the frame to be written
```

Author:
   Marvin Brown

System-Language:
   ADVISAR-FTN4

Routines Used:
   RMPAR, LUCON, IAND, CRSRC, LURQ, CLTTY, EXIT, WSCOM, MEMIO, ISHFT,
   MINO, MAXO, IFBRU

Core-Timing:

Examples:

PROGRAM PSHFT

Purpose: To shift an image within the current image screen.

Usage:

```
Under RTE-II or FMGR
     RU, PSHFT,,f
Under OS
     @PSHFT or *RU, PSHFT,,f
```

Where: f = the frame containing the image to be shifted.

Note:
- see DSCR for defining the current image screen.
- in OS, if @PSHFT is typed, the current frame which is being displayed on the RED signal system is used for f.
- the amount to shift in the x and y direction are prompted for
- both positive x and y indicate a shift down and right

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, LUCON, IAND, LURQ, CLTTY, RSCRN, IABS, MEMIO, IFBRK, EXIT

PROGRAM RW7

Purpose:
To rewind the 7-track tape drive and reset the current position.

Usage:
```
Under RTE-II or FMGR
     RU, RW7
Under OS
     @RW7
```

Author:
Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

WCUR7, EXIT

PROGRAM RW9

Purpose:

To rewind the 9-track tape drive and reset the current position.

Usage:

```
Under RTE-II or FMGR
     RU, RW9
Under OS
     @RW9
```

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

WCUR9, EXIT

PROGRAM SCOM

Purpose:
To read and write the ADVISAR system common area.

Usage:
Under RET II or FMGR:
RU, SCOM

Under OS:
@SCOM

Note:
read and write key, common location and value are prompted for.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RMPAR, LUCON, LURQ, CLTTY, RSCOM, WSCOM, EXIT

Example:

```
>@SCOM
>ENTER KEY (1=READ, 2=WRITE), LOCATION, VALUE - 1,5
>LOCATION 5 CONTAINS 0
>
```

PROGRAM SDEMO

Purpose:
This program transfers a frame from memory to a frame file on 9-track 1600 BPI tape.

Usage:

```
   Under RTEII or FMGR:
     RU, SDEMO,,f

   Under OS
     @SDEMO or *RU, SDEMO,,f

   Where:
     f = the frame to be transferred to mag tape.
```

Note:

- in OS, f is interperated as the current frame being displayed on the RED signal system.
- see DEMO for restoring a frame file.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RMPAR, IAND, MEMIO, BFOUT, EXIT

Example:

PROGRAM SKIP

Purpose:
To provide convenient mag tape manipulation.

Usage:
Under RTE II or FMGR:

```
RU, SKIP, fr, mt, n
```

Under OS:

```
*RU, SKIP, fr, mt, n
```

Where:
fr = F or R for files or records, respectively.
mt = the logical unit number of the mag tape.
n = the number of files to skip. (Negative is backwards, zero skips to the beginning of the file).

Errors:
If neither F or R are the first parameter, an error message is output.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
SKPFL, RMPAR, IABS, EXEC

Example:

PROGRAM T

Purpose:
To print the current x, y location of the center of the cursor on the screen.

Usage:
Under OS:
T or @T

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RMPAR, FRCUR, LURQ, CLTTY, EXIT

Example:

```
>T
>CURSOR POSITION (x, y) = 500, 217
>
```

PROGRAM TLIST

Purpose:
To print file and record information for a mag tape.

Usage:
Under RET II or FMGR:

```
RU, TLIST,,mt,nf,fg
```

Where:

```
mt = the mag tape logical unit
nf = the number of files to go through
fg = 0 then print all record information.
   ≠ 0 then print only the file information.
```

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RMPAR EXEC, ISUNT

Example:

PROGRAM T7

Purpose: To read an image from the 7 track tape drive in external format and store it in the current image screen (see DSCR) in the Intel memory.

Usage:

```
Under RTE-II or FMGR
  RU, T7
Under OS
  @T7
```

Note:
- file, records and elements are prompted for.
- if an end of file is encountered, the number of records in the file is output.
- if a negative file or record is entered, the T7 is aborted.

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, PICTR, EXIT, RCUR 7, SKPFL, BFINP, ISUNT, PTAPE, LENGT, RDREC, WCUR 7

PROGRAM T7HP

Purpose:

To read an image from the 7 track tape drive in HP format and store it in the current image screen (see DSCR) in the Intel memory.

Usage:

```
Under RTE-II or FMGR
  RU, T7HP
Under OS
  @T7HP
```

Note:
- file, records and elements are prompted for.
- if an end of file is encountered, the number of records in the file is output.
- if a negative file or record is entered, the T7HP is aborted.

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, PICTR, EXIT, RCUR7, SKPFL, BFINP, ISUNT, PTAPE, LENGT, RDREC, WCUR7

PROGRAM T9L

Purpose: To read an image from the 9 track tape drive at 800 BPI and store it in the current image screen (see DSCR) in the Intel memory.

Usage:

```
Under RTE-II or FMGR
  RU, T9L
Under OS
  @T9L
```

Note:
- file, records and elements are prompted for.
- if an end of file is encountered, the number of records in the file is output.
- if a negative file or record is entered, the T9L is aborted.

Author: Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, PICTR, EXIT, RCUR9, SKPFL, BFINP, ISUNT, PTAPE, LENGT, RDREC, WCUR9

PROGRAM T9H

Purpose: To read an image from the 9 track tape drive at 1600 BPI and store it in the current image screen (see DSCR) in the Intel memory.

Usage:

```
Under RTE-II or FMGR -
     RU, T9H
Under OS
     @T9H
```

Note:
- file, records and elements are prompted for
- if an end of file is encountered, the number of records in the file is output.
- if a negative file or record is entered, the T9H is aborted.

Author: Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RMPAR, PICTR, EXIT, RCUR9, SKPFL, BFINP, ISUNT, PTAPE, LENGT, RDREC, WCUR9

PROGRAM VIDIO

Purpose:
To change between the 480 and 512 vidio line display format. Vidio tapes can only be made in 480 line format.

Usage:
Under RTE II or FMGR:

```
RU, VIDIO
```

Under OS:

```
@VIDIO
```

Note:
VIDIO takes 3 seconds to run; this is for hardware stabilization.

Author:
Marvin Brown

System-Language:
ADVISAR - FTN4

Routines Used:
RSCOM, WSCOM, WAITS, EXIT

SUBROUTINE CLFRM

Purpose:
To set each location in a frame to a given value.

Usage:
CALL CLFRM (IF, IUAL)

Where:
IF = the frame number (0-7).
IVAL = the value to which the frame is set (0-255)

Author:
Marvin Brown

System-Language:
RTE II - ASMB

Routines Used:
EXEC, calls the program CLMEM

SUBROUTINE COLOR

Purpose: To set the value for plotting routines to write.

Usage: CALL COLOR (IVAL)

Where: IVAL = the eight bit value (0-255) for all plotting routines to write

Author: Mark Whitcomb

System-Language:
ADVISAR, FTN4

Routines Used:
WPCOM, entry point in PCOM

SUBROUTINE DASH

Purpose: To set the dash line mode for line and vector plotting routines

Usage: CALL DASH (IDASH)

Where: IDASH is a 4 word linear array containing the dash pattern

IDASH (1) = Number of bits on

IDASH (2) = Number of bits off

IDASH (3) = Number of bits on

IDASH (4) = Number of bits off

Note: to reset dash mode to fall lines, set IDASH (1) or IDASH (3) to any integer > zero and set IDASH (2) and IDASH (4) to zero.

Author: Mark Whitcomb

System-Language:
ADVISAR, FTN4

Routines Used:
WPCOM, an entry point in PCOM

SUBROUTINE ENHIO

Purpose:

To read or write the enhancement tables.

Usage:

```
CALL ENHIO (KEY, IS, IP, IBUF, LBUF)

Where:  KEY = 1 for read enhancement table
            = 2 for write enhancement table
         IS = signal system (0-4)
              0 = red, 1 = green, 2 = blue, 3 = B/W1, 4 = B/W2
         IP = first enhancement location to read or write
       IBUF = enhancement buffer (unpacked 8-bit bytes)
       LBUF = number of enhancement values
```

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

IAND, LURQ, EXEC, MOD

SUBROUTINE FRAME

Purpose: To set the frame number for all plotting routines to write to.

Usage: CALL FRAME (IFR)

Where: IFR = integer frame number (0-7)

Author: Mark Whitcomb

System-Language:

ADVISAR, FTN4

Routines Used:

WPCOM, an entry point in PCOM
IAND

SUBROUTINE FRMOV

Purpose: To transfer an entire frame to another frame

Usage: CALL FRMOV (IFR, IFW)

Where: IFR = frame to read from
IFW = frame to write to

Author: M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

EXEC, program MOVE

SUBROUTINE LINE

Purpose: To plot a line to a frame in the memories.

Usage: CALL LINE (IX1, IY1, IX2, IY2)

Where: IX1 = integer X coordinate of starting point (0-511)
IY1 = integer Y coordinate of starting point (0-511)
IX2 = integer X coordinate of ending point (0-511)
IY2 = integer Y coordinate of ending point (0-511)

Note: Out of range (0-511) values are legal, but only points lying between (0-511) are plotted. Actual line includes all points in between (IX1, IY1) and (IX2, IY2), including (IX2, IY2) but excluding (IX1, IY1).
Line is drawn conforming to line width, dash pattern and value as set by other routines (WDTH, DASH, COLOR). Default values are:
width: 1
dash: full line,
value: 255

The plot pointer is set to (IX2, IY2)

Author: Mark Whitcomb

System-Language:
ADVISAR, FTN4

Routines Used:
RPCOM, WPCOM, entry points in PCOM, IABS, SQRT, PSTRT, PQUIT, PLINE, PDASH

FUNCTION MASK (MASK)

Purpose:

To set refresh memory mask.

Usage:

```
CALL MASK (IMASK)
Where:  IMASK = 8-bit memory mask (0-255)
Note:  if IMASK = 0 then the current mask value is returned.
```

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

IAND

SUBROUTINE MEMIO

Purpose: To read and write an Intel memory frame.

Usage:

```
CALL MEMIO (KEY, IF, IROW, ICOL, IBUF, LBUF)
Where:  KEY = 1 for reading the memory
            = 2 for writing the memory
         IF = frame to read or write (0-7)
       IROW = row position to start transfer (0-511)
       ICOL = column position to start transfer (0-511)
       IBUF = data buffer (unpacked 8-bit bytes)
       LBUF = number of values to transfer
```

Note: Values are transferred incrementing the column counter then the row counter.

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

IAND, ISHFT, LURQ, EXEC

Subroutine MOV

Purpose: To move the plotting pointer (for Vector routines) to a specified point.

Usage: CALL MOV (IX, IY)

```
Where:  IX = integer  X  location in frame (0-511)
        IY = integer  Y  location in frame (0-511)
```

Note: a point lying outside the range (0-511) is legal, however, no points can be plotted, since it lies off the screen.

Author: Mark Whitcomb

System-Language:
ADVISAR, FTN4

Routines Used:
WPCOM, an entry point in PCOM

SUBROUTINE MOVR

Purpose: To set the plotting pointer (for vector routines) to its previous position plus relative corrdinates.

Usage: CALL MOVR (IX, IY)

Where: IX = integer increment in X direction
IY = integer increment in Y direction

Note: IX, IY can be any legal integers. Pointer is set exactly to (OLDX + IX, OLDY + IY). If resultant pointer is off screen, simply no points can be plotted there.

Author: Mark Whitcomb

System-Language:

ADVISAR, FTN4

Routines Used:

RPCOM, WPCOM, entry points in PCOM

SUBROUTINE MPNT

Purpose: To output a single point to the Intel memories.

Usage: CALL MPNT (IX, IY, IVAL)

```
Where:  IX = integer  X  frame coordinate (0-511)
        IY = integer  Y  frame coordinate (0-511)
      IVAL = 8 bit value to be written
```

Note: The lower 8 bits of IVAL are masked off to be written. If IX or IY are out of the range (0-511), no point is written. Input MUX, output MUX and control buffer are assumed to be set properly.

This routine shuts off the interrupt system to do the output, waiting for its completion.

Author: Mark Whitcomb

System-Language:

ADVISAR, ASMB

Routines Used:

.ENTR, $LIBR, $LIBX

PCOM

Purpose: To allow direct access to the plot common area

Usage: Entry Points: WPCOM: write plot common
RPCOM (IWORD, IDAT) or CALL RPCOM (IWORD, IDAT)

Where: IWORD = word number in common area
IDAT = data to be written or read

Word Arrangement:
Word 0 = frame number
Word 1 = current X pointer position
Word 2 = current Y pointer position
Word 3 = dash mode flag
Word 4 = dash code 1 (on 1)
Word 5 = dash code 2 (off 1)
Word 6 = dash code 3 (on 2)
Word 7 = dash code 4 (off 2)
Word 8 = color (0-255)
Word 10 = current dash position
Word 11 = current dash count

Author: Mark Whitcomb

System-Language:

ADVISAR, ASMB

Routines Used:

.ENTR

SUBROUTINE PDASH

Purpose: To plot a single width dashed line to a frame in the memories.

Usage: CALL PDASH (IX1, IY1, IX2, IY2, IDASH, ID, KOUNT IVAL)

```
Where:  IX1 = starting  X  location
        IY1 = starting  Y  location
        IX2 = ending  X  location
        IY2 = ending  Y  location
      IDASH = 4 word array with dash pattern
         ID = current subscript in IDASH
      KOUNT = current dash count
       IVAL = eight bit value to be written
```

Note: X, Y locations off screen are ignored. A dashed line is drawn to include (IX2, IY2) but to exclude (IX1, IY1).

Assumption is made that memory input MUX, control buffer, etc. are properly set.

Author: Mark Whitcomb

System-Language:
ADVISAR, FTN4

Routines Used:
IABS, ABS, SQRT, MPNT

SUBROUTINE PLINE

Purpose: To plot a single width line to a frame in the memories.

Usage: CALL PLINE (IX1, IY1, IX2, IY2, IVAL)

```
Where:  IX1 = starting  X  location
        IY1 = starting  Y  location
        IX2 = ending  X  location
        IY2 = ending  Y  location
       IVAL = eight bit value to be written
```

Note: X, Y locations off screen are ignored. Line is drawn from (IX1, IY1) to (IX2, IY2) including the point (IX2, IY2), but excluding the point (IX1, IY1).

Assumption is made that memory input MUX, control buffer, etc. are properly set.

Author: Mark Whitcomb

System-Language:

ADVISAR, FTN4

Routines Used:

IABS, MPNT

SUBROUTINE PNT

Purpose: To plot a single point to a frame in the memories.

Usage: CALL PNT (IX, IY)

Where: IX = integer X coordinate
IY = integer Y coordinate

Note: if IX or IY is off screen, no point is written. The plot pointer is always set to (IX, IY).

Author: Mark Whitcomb

System-Language:
ADVISAR - FTN4

Routines Used:
RPCOM, WPCOM, entry points in PCOM, MPNT, PSTRT, PQUIT

SUBROUTINE PQUIT

Purpose: To reset the memories after plotting.

Usage: CALL PQUIT

Author: Mark Whitcomb

System-Language:
ADVISAR - FTN4

Routines Used:
EXEC, LURQ

SUBROUTINE PSTRT

Purpose: To set up the memories for plotting

Usage: CALL PSTRT (IFR)

Where: IFR = frame number to plot to (0-7)

Author: Mark Whitcomb

System-Language:
ADVISAR - FTN4

Routines Used:
EXEC, LURQ

SUBROUTINE PSYM

Purpose: To output a dot matrix (characters) to a frame in the memories.

Usage:

CALL PSYM (IX, IY, MATC, NC, NR, ISIZE, IORNT, IVAL)

Where: IX = integer X location of matrix upper left hand corner
IY = integer Y location of matrix upper left hand corner
MATC = linear array with bit pattern of dot matrix
NC = number of columns in dot matrix (dimension of MATC)
NR = number of rows in dot matrix (number of bits in one word - right justified, must be less than or equal to 16)
ISIZE = character size
1 = matrix size
2 = twice matrix size
3 = thrice matrix size, etc.
IORNT = character orientation
1 = horizontal, 0 degrees
2 = vertical, 90 degrees
3 = horizontal, 180 degrees
4 = vertical, 270 degrees
IVAL = 8 bit value to be written

Author: Mark Whitcomb

System-Language:

ADVISAR - FTN4

Routines Used:

IAND, ISHFT, MPNT

SUBROUTINE PWRT

Purpose: To write a string of alphanumeric dot matrix characters to the memories.

Usage:

CALL PWRT (IX, IY, ICHAR, NCHAR, ISIZE, IORNT)

Where: IX = integer X location of upper left hand corner of first character
IY = integer Y location of upper left hand corner of first character
ICHAR = linear array of packed ASCII characters (Hollerith format)
NCHAR = number of characters in ICHAR
ISIZE = character size
1 = 5 x 7 dot matrix (8 x 8 with spacing)
2 = 10 x 14 dot matrix (16 x 16 with spacing)
3 = 15 x 21 dot matrix (24 x 24 with spacing), etc.
IORNT = character orientation
1 = horizontal, increasing X (0°)
2 = vertical, decreasing y (90°)
3 = horizontal, decreasing X (180°)
4 = vertical, increasing Y (270°)

Author: Mark Whitcomb

System-Language:

ADVISAR - FTN4

Routines Used:

RPCOM, an entry point in PCOM
IAND, ISHFT, PSTRT, PQUIT, PSYM

SUBROUTINE RCURP

Purpose: To read the current cursor position.

Usage: CALL RCURP (IX, IY)

```
Where:  IX = current  X  position of center of cursor returned (0-511)
        IY = current  Y  position of center of cursor returned (0-511)
```

Author: M. Brown

System-Language:
ADVISAR-FTN4

Routines Used:
RSCOM

SUBROUTINE RCURS

Purpose: To read the current cursor dimensions

Usage:

```
CALL RCURS (IXD, IYD)

Where: IXD = current  X  dimension of cursor returned (0-511)
       IYD = current  Y  dimension of cursor returned (0-511)
```

Author:

M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RSCOM

SUBROUTINE RCURT

Purpose: To read the cursor type and color parameters

Usage:

```
CALL RCURT (ITYPE, ICOLR, IBW)

Where:  ITYPE = cursor type key returned * (0-8)
        ICOLR = cursor color returned** (0-8)
          IBW = black and white cursor returned *** (0-4)

* cursor types are:  0 = no cursor
                     1 = solid box
                     2 = box outline
                     3 = window
                     4 = windpw with box
                     5 = cross-hairs
                     6 = four corner dots
                     7 = window with cross-hairs
                     8 = window with fourcorner dots

**cursor colors:     0 = flashing    5 = blue
                     1 = black       6 = purple
                     2 = red         7 = turquoise
                     3 = green       8 = white
                     4 = yellow
(more on back)
```

Author:
M. Brown

System-Language:
ADVISAR-FTN4

Routines Used:
RSCOM

***black-white key: 0 = flashing
1 = black on both monitors
2 = black on first, white on second
3 = white on first, black on second
4 = white on both

SUBROUTINE RDISP

Purpose: To read the display parameter associated with each signal system.

Usage:

```
DIMENSION ISS (5)
.
.
.
CALL RDISP (ISS)
```

Where: ISS = array containing display parameters returned

And: ISS(I) $\geq$ 0; value is frame currently being displayed on that signal system (0-7)
ISS(I) < 0; indicates that the signal system is off

Author: M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

IAND, RSCOM

SUBROUTINE RJOY1

Purpose: To read the current position of joystick one.

Usage: CALL RJOY1 (IX, IY)

```
Where:  IX = X  position of joystick one returned (0-511)
        IY = Y  position of joystick one returned (0-511)
```

Author: M. Brown

System-Language:

ADVISAR-FTN4

Routines Used:

RSCOM

SUBROUTINE RJOY2

Purpose: To read the current position of joystick two.

Usage: CALL RJOY2 (IX, IY)

```
Where:  IX = X  position of joystick two returned (0-511)
        IY = Y  position of joystick two returned (0-511)
```

Author: M. Brown

System-Language:

ADVISAR-FTN4

Routines Used:

RSCOM

SUBROUTINE RLOOP

Purpose: To read the loop parameters associated with a signal system.

Usage:

```
DIMENSION LOOP (12)
.
.
.
CALL RLOOP (ISS, LOOP)

Where:  ISS = signal system (0-4)
       LOOP = array containing loop parameters returned as follows:
              LOOP(1) = loop key*
                  (2) = loop frame count (1-8)
                  (3) = loop rate (≥1; # of frame interrupts)
                  (4) = loop first frame dwell(≥1; # of frame interrupts)
                  (5) = first frame in loop (0-7)
                  (6) = second frame in loop(0-7)
                   .
                   .
                   .
                 (12) = last frame in loop

       * loop key = -3 reset
                  = -2 back
                  = -1 advance
                  = 0 hold
                  = 1 loop
```

Author:

M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

MOD, RSCOM

SUBROUTINE ROURL

Purpose: To read the number of overlays and the overlay frames associated with a signal system.

Usage:

```
DIMENSION IOVS(3)
.
.
.
CALL ROURL (ISS, N, IOVS)

Where:  ISS = signal system (0-4)
          N = number of overlays returned (0-3)
       IOVS = array containing overlay frame numbers returned
```

Author: M. Brown

System-Language:

ADVISAR, FTN4

Routines Used:

MOD, IAND, ISHFT, RSCOM

SUBROUTINE RPOTS

Purpose: To read the output of the joystick potentiometers.

Usage:

```
DIMENSION IPOTS (4)
.
.
.
CALL RPOTS (IPOTS)
```

Where: IPOTS = array of outputs from potentiometers 1-4 returned (0-2047)

Author: M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RSCOM

SUBROUTINE RSCRN

Purpose: To read the current image area on the screen

Usage:

```
CALL RSCRN (IBX, IEX, IBY, IEY)

Where:  IBX = beginning X screen coordinate
        IEX = ending X screen coordinate
        IBY = beginning Y screen coordinate
        IEY = ending Y screen coordinate
```

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RSCOM

SUBROUTINE RVIDO

Purpose: To read the video system key

Usage: CALL RVIDO (IVIDO)

```
Where:  IVIDO = video system key returned (0-1)
            0 = 512 positions
            1 = 480 positions
```

Author: M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

IAND, RSCOM

SUBROUTINE VCT

Purpose: to draw a vector on a frame in the memories.

Usage: CALL VCT (IX, IY)

```
Where IX = integer  X  location of end point
      IY = integer  Y  location of end point
```

Note: A line is drawn from the old plotting pointer position to (IX, IY). The pointer is set to (IX, IY). The line is drawn with width, dash pattern and value as specified by other routines.

Author: Mark Whitcomb

System-Language:

ADVISAR - FTN4

Routines Used:

RPCOM, an entry point in PCOM
LINE

SUBROUTINE VCTR

Purpose: To draw a vector to a frame in the memories, using relative coordinates.

Usage: CALL VCTR (IX, IY)

Where: IX = integer X increment
IY = integer Y increment

Note: A line is drawn from the old plotting pointer to the location of the older pointer plus (IX, IY). The pointer is then set to its old value plus (IX, IY).

The line is drawn with width, value and dash pattern as set by other routines.

Author: Mark Whitcomb

System-Language:

ADVISAR - FTN4

Routines Used:

RPCOM, an entry point in PCOM
LINE

SUBROUTINE WAITS

Purpose: To suspend a program for a given number of video retraces.

Usage: CALL WAITS (N)

```
Where:  N > 0 then the program is suspended for N/30 seconds.
          < 0 then the program is suspended for -N/60 seconds.
          = 0 then the program is suspended until the next video
              retrace.
```

Author: Marvin Brown

System-Language:

ADVISAR - ASMB

Routines Used:

:ENTR, RNRQ, .WWAT

SUBROUTINE WCURP

Purpose:
To write out the desired cursor position.

Usage:
CALL WCURP (IX, IY)

```
Where:  IX = desired  X  position of cursor center (0-511)
        IY = desired  Y  position of cursor center (0-511)
```

Author:
M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

WSCOM, WAITS

SUBROUTINE WCURS

Purpose: To write out the desired cursor dimensions.

Usage: CALL WCURS (IXD, IYD)*

Where: IXD = desired X dimension of cursor size (0-511)
IYD = desired Y dimension of cursor size (0-511)

*Note: The cursor will be turned off following execution of this routine.

Author: M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

WSCOM, WAITS

SUBROUTINE WCURT

Purpose: To write out the desired cursor type and color parameters.

Usage: CALL WCURT (ITYPE, ICOLR, IBW)

Where: ITYPE = cursor type * (0-8)
ICOLR = cursor color ** (0-8)
IBW = black-white cursor key *** (0-4)

* cursor types:
0 = no cursor
1 = solid box
2 = box outline
3 = window
4 = window with box
5 = cross hairs
6 = four corner dots
7 = window with cross-hairs
8 = window with four corner dots

**cursor colors:
0 = flashing
1 = black
2 = red
3 = green
4 = yellow
5 = blue
6 = purple
7 = turquoise
8 = white

(more on back)

Author:

M. BROWN

System-Language:

ADVISAR-FTN4

Routines Used:

WSCOM

*** black-white key: 0 = flashing

1 = black on both monitors

2 = black on first, white on second

3 = white on first, black on second

4 = white on both

SUBROUTINE WDISP

Purpose: To write the frame number to be displayed on each signal system, optionally to turn the signal system on or off.

Usage:
```
DIMENSION ISS (5)
.
.
.
CALL WDISP (ISS)
```

Where ISS = array containing frame numbers to be displayed on each signal system (0-7)

And ISS(I) $\geq$ 0; value is frame number to be displayed, signal system should be turned on

ISS(I) < 0; turn the signal system off

Author:

M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RSCOM, IAND, IOR, WSCOM

SUBROUTINE WDTH

Purpose: To set the width for lines to be drawn.

Usage:

```
CALL WDTH (IW)

Where:  IW = integer width of line (in points of a horizontal or
             vertical line)
```

Author:

Mark Whitcomb

System-Language:

ADVISAR - FTN4

Routines Used:

WPCOM, an entry point in PCOM

SUBROUTINE WLOOP

Purpose: To write the loop parameters for a signal system.

Usage:

```
DIMENSION LOOP (12)
.
.
.
CALL WLOOP (ISS, LOOP)

Where:  ISS = signal system (0-4)
       LOOP = array containing loop parameters to write

LOOP (1) = loop key*
     (2) = loop frame count (1-8)
     (3) = loop rate (≥1; # of frame interrupts)
     (4) = loop first frame dwell (≥1; # of frame interrupts)
     (5) = first frame in loop (0-7)
     (6) = second frame in loop (0-7)
      .
      .
      .
     (12)= last frame in loop
* loop key = -3 reset
           = -2 back
           = -1 advance
           = 0 hold
           = 1 loop
```

Author:

M. Brown

System-Language:

ADVISAR - FTN4

Routines Used:

MOD, WSCOM

SUBROUTINE WSCRN

Purpose: To set the current image area on the screen.

Usage:

```
CALL WSCRN (IBX, IEX, IBY, IEY)

Where - IBX = beginning X screen coordinate
        IEX = ending X screen coordinate
        IBY = beginning Y screen coordinate
        IEY = ending Y screen coordinate
```

Author:

Marvin Brown

System-Language:

ADVISAR - FTN4

Routines Used:

RSCOM

SUBROUTINE WVIDO

Purpose: To write out the desired video system key

Usage:

```
CALL WVIDO (IVIDO)

Where:  IVIDO = video system key (0-1)
            0 = 512 positions
            1 = 480 positions
```

Author:

M. Brown

System-Language:

ADVISAR-FTN4

Routines Used:

RVIDO, IAND, WAITS, WSCOM

Appendix E

List of participating scientific personnel and advanced degrees earned:

Marvin Brown

Thomas Brubaker

Robert Fitch -- MS in Electrical Engineering

Uwe Haas

Kenneth R. Morris

Andrew Negri

David Reynolds

Eric Smith -- MS in Atmospheric Science

Gregory Smith -- MS in Electrical Engineering

Richard Stodt

Thomas VonderHaar

Mark Whitcomb

DATE

ILME